

TUGAS AKHIR - KI141502

**PEMERINGKATAN PERGURUAN TINGGI MENGGUNAKAN
METODE *LDA-ADABOOST.MH* UNTUK MENGUKUR
TINGKAT KESIAPTERAPAN TEKNOLOGI PERGURUAN
TINGGI DI INDONESIA**

**Arga Lancana Yuananda
NRP 5113 100 057**

**Dosen Pembimbing I
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D**

**Dosen Pembimbing II
Bagus Setya Rintyarna, S.T., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

**PEMERINGKATAN PERGURUAN TINGGI MENGGUNAKAN
METODE *LDA-ADABOOST.MH* UNTUK MENGUKUR
TINGKAT KESIAPTERAPAN TEKNOLOGI PERGURUAN
TINGGI DI INDONESIA**

**Arga Lancana Yuananda
NRP 5113 100 057**

**Dosen Pembimbing I
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.**

**Dosen Pembimbing II
Bagus Setya Rintyarna, S.T., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

**UNIVERSITY RANKING USING *LDA-ADABOOST.MH*
METHOD TO MEASURE TECHNOLOGY READINESS
LEVEL OF UNIVERSITY IN INDONESIA**

**Arga Lancana Yuananda
NRP 5113 100 057**

**Supervisor I
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.**

**Supervisor II
Bagus Setya Rintyarna, S.T., M.Kom.**

**INFORMATICS DEPARTMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PEMERINGKATAN PERGURUAN TINGGI MENGUNAKAN METODE *LDA-ADABOOST.MH* UNTUK MENGUKUR TINGKAT KESIAPTERAPAN TEKNOLOGI PERGURUAN TINGGI DI INDONESIA

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Manajemen Informasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh
ARGA LANCANA YUANANDA
NRP. 5113 100 057

Disetujui oleh Dosen Pembimbing Tugas Akhir.

Prof. Drs. Ec. Ir. RIYANARTO S.E.P.M.
M.Sc., Ph.D.
NIP: 19590803 198601 1 001



(pembimbing 1)

BAGUS SETYA RINTYARNA S.T.
M.Kom.
NIK: 0509502

(pembimbing 2)

SURABAYA
JUNI, 2017

[Halaman ini sengaja dikosongkan]

PEMERINGKATAN PERGURUAN TINGGI MENGUNAKAN METODE *LDA- ADABOOST.MH* UNTUK MENGUKUR TINGKAT KESIAPTERAPAN TEKNOLOGI PERGURUAN TINGGI DI INDONESIA

Nama	: Arga Lancana Yuananda
NRP	: 5113100057
Jurusan	: Teknik Informatika – FTIf ITS
Dosen Pembimbing I	: Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc.,Ph.D.
Dosen Pembimbing II	: Bagus Setya Rintyarna, S.T., M.Kom.

Abstrak

Perihal ketatnya persaingan antar perguruan tinggi, terdapat beberapa cara untuk memberikan suatu ukuran kualitas perguruan tinggi, salah satunya yaitu Pemeringkatan Perguruan Tinggi berdasarkan tingkat kesiapterapan teknologi.

Tingkat kesiapterapan teknologi perguruan tinggi secara umum merupakan tingkat kematangan hasil penelitian/pengembangan teknologi masing-masing perguruan tinggi dengan tujuan untuk dapat diadopsi oleh pengguna, baik untuk pemerintah, industri maupun masyarakat. Hal ini tentu saja dapat dijadikan suatu acuan dalam Pemeringkatan Perguruan Tinggi khususnya di Indonesia.

Terdapat dua inti kriteria penilaian untuk menghasilkan Pemeringkatan Perguruan Tinggi, yaitu penilaian kualitatif dan kuantitatif. Penilaian kualitatif didapatkan melalui survei dari perwakilan masing-masing perguruan tinggi sedangkan kuantitatif didapatkan melalui data konkret yang sudah disediakan oleh masing-masing perguruan tinggi.

*Dalam Tugas Akhir ini diusulkan sebuah gagasan baru untuk sistem pemeringkatan Perguruan Tinggi di Indonesia menggunakan metode teks mining *LDA-AdaBoost.MH* sebagai*

pengganti penilaian kualitatif sebelumnya yang menggunakan survei secara manual kepada masing-masing perwakilan perguruan tinggi. Cara kerja metode LDA-AdaBoost.MH ini adalah dengan menggali topik utama secara otomatis dari dokumen penelitian akademisi yang kemudian akan dihasilkan suatu nilai masing-masing perguruan tinggi berdasarkan standar Tingkat Kesiapterapan Teknologi di Indonesia.

Hasil implementasi sistem pemeringkatan Perguruan Tinggi menunjukkan bahwa kebenaran sistem telah mendekati ground truth yaitu pemeringkatan QS University Rankings dengan tingkat akurasi 86,4%, dengan hasil ini menunjukkan bahwa sistem telah berhasil melakukan optimasi kriteria kualitatif Reputasi Akademik secara lebih efisien dan menghemat waktu kerja.

Kata kunci: Kualitas Perguruan Tinggi, Kualitatif, Kuantitatif, Latent Dirichlet Allocation, LDA-AdaBoost.MH, Adaboost, Optimasi, Pemeringkatan Perguruan Tinggi, QS University Peringkat, Reputasi

UNIVERSITY RANKING USING LDA- ADABOOST.MH METHOD TO MEASURE TECHNOLOGY READINESS LEVEL OF UNIVERSITIES IN INDONESIA

Student Name : Arga Lancana Yuananda
NRP : 5113100057
Major : Informatics Department – FTIf ITS
Supervisor I : Prof. Drs. Ec. Ir. Rinyarto Sarno, M.Sc., Ph.D.
Supervisor II : Bagus Setya Rintyarna, S.T., M.Kom.

Abstract

Regarding the intense competition among universities, there are several ways to provide a quality measurement. One of which is University Ranking based on the technology readiness level.

The level of technological readiness in general is the maturity level of research / technological development of each university in order to be adopted by users, government, industry and society.

There are two core criteria for producing University Ranking, namely qualitative and quantitative criterias. Qualitative crieria obtained from the lecturer and employee survey of each college. Quantitative criteria obtained from data that already provided by each college.

This Final Project proposed a new concept for the University Ranking in Indonesia by using LDA-AdaBoost.MH mining method in lieu of the previous qualitative criteria using manual survey. LDA-AdaBoost.MH works by mining the main topic of academic journals automatically to get the qualitative criteria value based on the technology readiness level.

The result of University Ranking system implementation shows that it could compete the ground truth that is QS University Rankings with 86,4% accuracy level. This result indicates that the

system has succeeded in optimizing the qualitative criteria of Academic Reputation efficiently.

Keywords: Higher Education Quality, Qualitative, Quantitative, Latent Dirichlet Allocation, LDA-AdaBoost.MH, Adaboost, Optimization, Higher Education, QS University Ranking, Reputation

KATA PENGANTAR

Puji syukur kehadiran Allah SWT, karena atas berkat rahmat-Nya, penulis dapat menyelesaikan Tugas Akhir yang berjudul:

“Pemeringkatan Perguruan Tinggi Menggunakan Metode LDA-AdaBoost.MH Untuk Mengukur Tingkat Kesiapterapan Teknologi Perguruan Tinggi Di Indonesia”

Pengerjaan Tugas Akhir ini menjadi sebuah sarana untuk penulis memperdalam ilmu yang telah didapatkan selama menempuh pendidikan di kampus perjuangan Institut Teknologi Sepuluh Nopember Surabaya, khususnya dalam disiplin ilmu Teknik Informatika. terselesaikannya buku Tugas Akhir ini tidak terlepas dari bantuan dan dukungan semua pihak. Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Bapak, Ibu, adik dan keluarga yang selalu memberikan dukungan penuh untuk menyelesaikan Tugas Akhir ini.
2. Bapak Riyanarto Sarno dan Bapak Bagus Setya Rintyarna selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan Tugas Akhir ini.
3. Bapak dan Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
4. Seluruh staf dan karyawan FTIf ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.
5. Segenap dosen rumpun mata kuliah Manajemen Informasi.
6. Rekan-rekan mahasiswa S2 Teknik Informatika yang telah banyak meluangkan waktu dan bersedia membantu penulis dalam memahami pengerjaan Tugas Akhir ini.
7. Rekan-rekan pengerjaan satu tim: Donny Aliyanto, Anugerah Yulindra Setyaji, Eko Putro Fitrianto dalam penyelesaian Tugas Akhir.

8. Teman-teman seperjuangan anak didik Tugas Akhir Prof. Riyanarto Sarno yaitu Ahmad Zaenal Mustofa, Renanda Agustiantoro, Faisal Yanuar, Faisal Anugerah, Andi Putra, Hariyanto, Aldhiaz Fathra, Arfian Fidiantoro, dan M. Fariz Ponighzwa.
9. Teman-teman Teknik Informatika Angkatan 2013 yang selalu mendukung, menyemangati, membantu dan mendengarkan suka duka selama proses pengerjaan Tugas Akhir.
10. Teman-teman SMA yang melabeli diri mereka „Kita Keren“ secara sarkas.
11. Serta semua pihak yang turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Juni 2017

Arga Lancana Yuananda

DAFTAR ISI

LEMBAR PENGESAHAN	vii
Abstrak	ix
<i>Abstract</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR.....	xxi
DAFTAR TABEL	xxiii
DAFTAR KODE SUMBER.....	xxvii
1 BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Permasalahan.....	3
1.3 Batasan Permasalahan	4
1.4 Tujuan	4
1.5 Manfaat.....	5
1.6 Metodologi	6
1.6.1 Studi literatur	6
1.6.2 Analisis dan Perancangan Sistem.....	6
1.6.3 Implementasi Sistem.....	7
1.6.4 Uji Coba dan Evaluasi	7
1.6.5 Penyusunan Buku Tugas Akhir	8
1.7 Sistematika Penulisan.....	8
2 BAB II DASAR TEORI.....	11

2.1	University Rankings.....	11
2.2	<i>QS World University Rankings</i>	11
2.3	Teks Mining	19
2.4	Teknik Grabbing	19
2.4.1	Fungsi cURL.....	20
2.4.2	Fungsi File_get_contents.....	21
2.4.3	Simple HTML Dom Parser.....	21
2.4.4	Regular Expression (ReGex)	22
2.5	Pusat Data <i>Scopus</i>	22
2.6	<i>Google Scholar</i>	23
2.7	<i>Science and Technology Index (SINTA)</i>	24
2.8	Pangkalan Data Pendidikan Tinggi (PDDIKTI)	25
2.9	<i>Alexa Traffict Rank</i>	26
2.10	<i>MOZ Analytics</i>	27
2.11	Bahasa Pemrograman Python.....	27
2.12	Library Python	28
2.13	PHP dan HTML	32
2.14	<i>Database Management System (DBMS) MySQL</i>	33
2.15	<i>Part-Of-Speech Tagger (POS Tagger)</i>	33
2.16	<i>Bloom Taxonomy</i>	35
2.17	Metode <i>LDA-AdaBoost.MH</i>	37
2.18	Macam-macam Pengembangan Metode LDA	38
2.19	<i>Term Frequency – Invers Document Frequency (TF-IDF)</i>	39
2.19.1	Term Frequency (TF)	39
2.19.2	Invers Documen Frequency (IDF)	40

2.20 Fase Teks Mining (Metode LDA-AdaBoost.MH) Studi Kasus Pemeringkatan Perguruan Tinggi	40
2.21 Akreditasi Perguruan Tinggi	44
2.22 <i>Tingkat Kesiapterapan Teknologi (TKT)</i>	46
3 BAB III ANALISIS DAN PERANCANGAN SISTEM.....	49
3.1. Analisis.....	49
3.1.1 Cakupan permasalahan	49
3.1.2 Deskripsi umum sistem.....	49
3.1.3 Aktor	50
3.1.4 Spesifikasi Kebutuhan Sistem.....	50
3.1.5 Kasus Penggunaan	51
3.1.6 Conceptual Data Model (CDM)	62
3.1.7 Physical Data Model (PDM)	63
3.1.8 Label Kelas dalam Teks Mining Metode LDA- AdaBoost.MH.....	64
3.1.9 Ground Truth Pengujian Sistem.....	65
3.1.10 Dataset Pengujian Sistem.....	69
3.2 Tahap Perancangan	73
3.2.1 Perancangan Basis Data.....	73
3.2.2 Perancangan Antarmuka	83
3.2.3 Perancangan Kriteria Penilaian Perguruan Tinggi	86
3.2.4 Proses Sistem Aplikasi Pengguna (Diagram Aplikasi)	91
3.2.5 Proses Kerja Sistem Secara Keseluruhan (Flowchart Sistem)	92
3.2.6 Proses Teks Mining Metode LDA-AdaBoost (Flowchart Metode LDA-AdaBoost.MH).....	93

4	BAB IV IMPLEMENTASI SISTEM	94
4.1	Lingkungan Pengembangan Sistem	94
4.2	Implementasi Program pada Pemingkatan Universitas di Indonesia	94
4.2.1	Preprosesing.....	95
4.2.2	Pembaruan Korpus	95
4.2.3	Klasifikasi Term Frequency	95
4.2.4	Klasifikasi <i>LDA-AdaBoost.MH</i>	96
4.3	Implementasi Basis Data.....	96
4.4	Implementasi Antarmuka Pengguna	96
4.4.1	Antarmuka Melihat Halaman Utama	96
4.4.2	Antarmuka Hasil Pencarian Universitas	97
4.4.3	Antarmuka Detail Informasi Universitas	97
4.4.4	Antarmuka Hasil Pemingkatan Universitas	97
5	BAB V PENGUJIAN DAN EVALUASI	99
5.1	Lingkungan Uji Coba.....	99
5.2	Data Uji Coba.....	99
5.2.1	Data Korpus Tingkat Kesiaapterapan Teknologi	99
5.2.2	Data Abstrak Jurnal Akademik.....	102
5.2.3	Data <i>Science and Technology Index</i> (SINTA)	102
5.2.4	Data Jumlah Mahasiswa dan Dosen.....	103
5.2.5	Pengujian Fungsionalitas.....	103
5.3	Skenario Pengujian.....	111
5.3.1	Preprosesing.....	111
5.3.2	Pembaruan Korpus	112
5.3.3	Klasifikasi Term Frequency	113

5.3.4	Pembobotan Final Pemeringkatan	115
5.4	Evaluasi Pengujian	121
6	BAB VI KESIMPULAN DAN SARAN.....	123
6.1	Kesimpulan.....	123
6.2	Saran.....	124
	DAFTAR PUSTAKA.....	125
	LAMPIRAN	131
	Kode Sumber	131
	Gambar	165
	BIODATA PENULIS	171

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.3-1 Proses Utama Teks Mining	19
Gambar 2.9.1 Proses Klasifikasi <i>LDA-AdaBoost.MH</i>	37
Gambar 2.22-1 Hirarki Tingkat Kesiapterapan Teknologi (TKT).....	47
Gambar 1-1 Diagram Kasus Penggunaan.....	52
Gambar 1-2 Diagram Aktifitas Use Case UC-001	55
Gambar 1-3 Diagram Aktifitas Use Case UC-002	57
Gambar 1-4 Diagram Aktifitas Use Case UC-003	58
Gambar 1-5 Diagram Aktifitas Use Case UC-004	59
Gambar 1-6 Diagram Aktifitas Use Case UC-005	61
Gambar 1-7 Conceptual Data Model.....	62
Gambar 1-8 Physical Data Model	63
Gambar 3.2-1 Alur Menentukan Reputasi Akademik	86
Gambar 3.2-2 Alur Menentukan Skor Akreditasi	87
Gambar 3.2-3 Alur Menentukan Skor Rasio Mahasiswa dan Tenaga Penagajar Aktif	88
Gambar 3.2-4 Alur Menentukan Skor Jumlah Sitasi Akademisi Penelitian.....	89
Gambar 3.2-5 Diagram Aplikasi	91
Gambar 3.2-6 Flowchart Sistem Keseluruhan.....	92
Gambar 3.2-7 Flowchart Metode LDA-AdaBoost.MH..	93
Gambar A.1 Halaman Utama Bagian Pencarian	165
Gambar A.2 Halaman Utama Bagian Informasi Pemeringkatan	165
Gambar A.3 Halaman Utama Bagian Metode Penilaian	166
Gambar A.4 Halaman Utama Bagian Peta (Peringkat berdasarkan Region).....	166
Gambar A.5 Halaman Utama Bagian Lihat Peringkat dan <i>tab</i> Custom (untuk pembobotan sendiri)	167
Gambar A.6 Halaman Utama Bagian Tentang Kami ...	167
Gambar A.7 Halaman Hasil Pencarian.....	168
Gambar A.8 Halaman Detail Informasi Universitas	168
Gambar A.9 Halaman Hasil Pemeringkatan	169

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.15-1 Struktur Kelas kata POS Tagging	34
Tabel 1-1 Daftar Kebutuhan Fungsional Sistem	50
Tabel 1-2 Daftar Kebutuhan Fungsional Sistem	52
Tabel 1-3 Case Memeasukkan Presentase Bobot Kriteria Penilaian Pemeringkatan	53
Tabel 1-4 Use Case Melihat Peringkat Perguruan Tinggi Berdasarkan Tipe Pemeringkatan.....	56
Tabel 1-5 Use Case Mencari Perguruan Tinggi Berdasarkan Potongan Nama	57
Tabel 1-6 Use Case Mencari Perguruan Tinggi Berdasarkan Region Wilayah Provinsi.....	58
Tabel 1-7 Use Case Melihat Detail Informasi Utama Perguruan Tinggi.....	59
Tabel 1-8 Penjelasan Setiap Kelas Label	64
Tabel 1-9 Peringkat QS World University Rankings 2015-2016.....	65
Tabel 1-10 Peringkat Reputasi Akademik QS 2015-2016	66
Tabel 1-11 Peringkat QS World University Rankings 2016-2017.....	66
Tabel 1-12 Peringkat Reputasi Akademik QS 2016-2017	67
Tabel 1-13 Peringkat QS World University Rankings 2017-2018.....	67
Tabel 1-14 Peringkat Reputasi Akademik QS 2017-2018	68
Tabel 1-15 Contoh Hasil Analisa Manual Sebagai Ground Truth Reputasi Akademik	69
Tabel 1-16 Daftar Rincian Jumlah Kata Dataset Abstrak Paper Perguruan Tinggi.....	70
Tabel 1-17 Daftar Rincian Jumlah Kata Corpus Taxonomy Bloom.....	71
Tabel 1-18 Perbandingan Jumlah Kata Corpus TKT	72

Tabel 5.1 Tabel Pengujian Fitur Melihat Peringkat Universitas.....	103
Tabel 5.2 Tabel Pengujian Fitur Memberikan Bobot Kriteria	104
Tabel 5.3 Tabel Pengujian Fitur Mencari Universitas..	105
Tabel 5.4 Tabel Pengujian Fitur Mencari Informasi Peringkat Universitas Sesuai Provinsi.....	106
Tabel 5.5 Tabel Pengujian Fitur Melihat Detail Informasi Universitas.....	107
Tabel 5.6 Tabel Pengujian Fitur Mengelola Data Informasi Universitas	108
Tabel 5.7 Tabel Pengujian Fitur Melihat Metode Penilaian.....	110
Tabel 5.8 Hasil preprosesing	111
Tabel 5.9 Hasil Korpus Awal	112
Tabel 5.10 Hasil Pembaruan Korpus.....	113
Tabel 5.11 Hasil Pelabelan TKT awal.....	113
Tabel 5.12 Hasil Pelabelan TKT baru	114
Tabel 5.13 Hasil Probabilitas LDA terhadap TKT awal	115
Tabel 5.14 Hasil peringkat perguruan tinggi LDA terhadap TKT awal.....	115
Tabel 5.15 Hasil Pembobotan LDA terhadap TKT baru	116
Tabel 5.16 Hasil Peringkat Perguruan Tinggi LDA terhadap TKT baru	117
Tabel 5.17 Hasil Probabilitas LDA-AdaBoost.MH terhadap TKT awal.....	117
Tabel 5.18 Hasil peringkat perguruan tinggi LDA-AdaBoost.MH terhadap TKT awal	118
Tabel 5.19 Hasil Pembobotan LDA-AdaBoost.MH terhadap TKT baru	119
Tabel 5.20 Hasil Peringkat Perguruan Tinggi LDA-AdaBoost.MH terhadap TKT baru	119

Tabel 5.21 Ringkasan Performa LDA dan LDA- AdaBoost.MH.....	120
Tabel 5.22 Rangkuman Hasil Pengujian	121

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 2.4-1 Syntax cURL dengan setting option	
url.....	20
Kode Sumber 2.4-2 Syntax Fungsi file_get_contents	21
Kode Sumber 2.4-3 Syntax Penggunaan Library Simple	
HTML Dom Parser.....	21
Kode Sumber A.1 Preprosesing	146
Kode Sumber A.2 Pembaruan Korpus	148
Kode Sumber A.3 Klasifikasi TF	150
Kode Sumber A.4 <i>LDA-AdaBoost.MH</i>	156
Kode Sumber A.5 Implementasi Basis Data	164

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pengerjaan Tugas Akhir, dan sistematika penulisan Tugas Akhir.

1.1 Latar Belakang

Perguruan tinggi merupakan sebuah institusi pendidikan penyelenggara pendidikan tinggi. Pendidikan tinggi menjadi hal penting karena mencerminkan kondisi intelektual masyarakat dari sebuah negara. Membutuhkan perhatian dan evaluasi lebih dari pemerintah untuk dapat menghasilkan sumberdaya manusia yang baik dari perguruan tinggi. Hal ini tentunya untuk menghasilkan generasi yang mempunyai intelektual dan keterampilan tinggi ketika bersaing di dunia kerja kelak (K.S. Reddy, En Xie, Qingqing Tang, 2016). Kualitas pendidikan tinggi yang baik menjadi prioritas dan pusat perhatian pendidikan internasional. Pencapaian tujuan strategis perguruan tinggi bergantung pada faktor konstekstual, yang meliputi kepemimpinan suatu negara, ideologi negara, finansial, sosial budaya, komunikasi dan bahasa, serta proses pengambilan keputusan dari pemerintahan. Selain itu perguruan tinggi juga didorong oleh norma yang terkait dengan lingkungan dan nilai pendidikan itu sendiri (K.S. Reddy, En Xie, Qingqing Tang, 2016).

Selain itu perguruan tinggi menjadi salah satu sarana yang efektif untuk mengembangkan ilmu pengetahuan dan teknologi dengan berbagai cara, salah satu cara dengan melakukan penelitian. Penelitian merupakan salah satu wadah kreatifitas dan pengembangan potensi akademik bagi mahasiswa. Selain itu penelitian diselenggarakan untuk membuat seorang akademisi dapat menghasilkan sebuah karya dalam dunia pendidikan (K.S. Reddy, En Xie, Qingqing Tang, 2016). Jumlah populasi masyarakat suatu negara yang banyak maka harus berbanding lurus

dengan pengembangan intelektual dan keterampilan generasi muda, sebagai investasi negara dalam dunia pendidikan.

Salah satu upaya untuk meningkatkan kualitas perguruan tinggi adalah dengan melihat hasil pemeringkatan perguruan tinggi di seluruh dunia (Tatiana Sidorenko, Tatiana Gorbatoва, 2015). Perguruan tinggi yang masuk dalam peringkat dunia akan menjadi acuan dan fokus bahwa kualitas pendidikan di negara tersebut telah bagus, dari berbagai segi kriteria. Ada banyak media publikasi pemeringkatan perguruan tinggi, salah satunya adalah publikasi tahunan *QS World University Rankings* (Mu-Hsuan Huang, 2012). Dalam pemeringkatan perguruan tinggi, tentunya ada banyak kriteria penilaian yang menjadikan perguruan tinggi tersebut berkualitas dan layak untuk menempati peringkat dalam daftar.

Tingkat kualitas pendidikan di Indonesia sejauh ini dapat dikatakan kurang baik. Indonesia masih berfokus pada jumlah partisipasi pendidikan, tidak pada pembangunan kualitas pendidikan. Berdasarkan posisi peringkat publikasi tahunan *QS World University Rankings*, perguruan tinggi di Indonesia masih berada pada urutan 67 regional Asia yang ditempati oleh Universitas Indonesia (UI). Berbeda jauh jika dibandingkan dengan negara tetangga, yaitu Singapura yang mempunyai jumlah penduduk lebih sedikit dari Indonesia berhasil berada pada peringkat 1 region Asia yang ditempati oleh National University of Singapore (NUS) (*QS World University Rankings*, 2016). Di Indonesia sendiri kualitas pendidikan yang menonjol hanya berada pada perguruan tinggi tertentu saja, hal ini yang menyebabkan kualitas pendidikan di Indonesia kurang merata.

Salah satu cara untuk mengetahui merata atau tidaknya kualitas pendidikan tinggi di Indonesia dengan melihat Tingkat Kesiapterapan Teknologi (TKT) dari perguruan tinggi. TKT menggambarkan tingkat kematangan perguruan tinggi dalam hal penelitian dan pengembangan teknologi (Kementerian, Riset Teknologi dan Pendidikan Tinggi, 2016). Tujuan utama dari penelitian secara umum sejalan dengan tujuan perguruan tinggi itu

sendiri, sebagai sarana yang efektif untuk mengembangkan ilmu pengetahuan dan pengembangan potensi sumberdaya manusia.

Kembali kepada publikasi tahunan *QS World University Rankings* terdapat berbagai macam kriteria penilaian yang secara umum menggambarkan kondisi perguruan tinggi (Mu-Hsuan Huang, 2012), yang mana dapat menjadi *value* untuk meningkatkan tingkat kesiapterapan (TKT) dari perguruan tinggi. Kriteria kualitatif dalam *QS World University Rankings* menjadi *value* yang paling berpengaruh dalam melihat kualitas perguruan tinggi. Kriteria ini didapatkan dengan proses manual yang teliti dan akurat terhadap hasil, sehingga membutuhkan waktu yang lama untuk menghasilkan pemeringkatan yang ideal (Mu-Hsuan Huang, 2012). Salah satu pilihan untuk mengoptimasi kriteria kualitatif adalah dengan menerapkan metode kecerdasan komputasional berupa metode teks mining *LDA-AdaBoost.MH* (Daniel Ramage, David Hall, Ramesh Nallapati, Christopher D. Manning, 2009).

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana mengumpulkan data masukan sebagai bahan penilaian kriteria perguruan tinggi?
2. Bagaimana menghasilkan skor akhir dari kriteria penilaian perguruan tinggi yang bersifat kualitatif menggunakan topic modelling metode *LDA-AdaBoost.MH*?
3. Bagaimana menghitung skor akhir dari data kriteria penilaian perguruan tinggi yang bersifat kuantitatif?
4. Bagaimana menentukan skor akhir secara keseluruhan untuk sebuah perguruan tinggi?

Bagaimana menganalisa hasil keluaran sistem dengan *Ground Truth QS University Rankings* yang sudah ditetapkan sehingga menghasilkan hasil akurasi?

1.3 Batasan Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan sebagai berikut:

1. Bahasa pemrograman yang digunakan adalah Python untuk proses *teks mining* dokumen abstrak *paper*, PHP dan HTML untuk membangun sistem *grabbing* dan membangun desain antarmuka.
2. Database management sistem (DBMS) yang digunakan adalah MySQL database.
3. Perguruan tinggi yang dinilai hanya perguruan tinggi dalam wilayah negara Indonesia.
4. Menggunakan acuan kriteria dari media publikasi tahunan *QS World University Rankings*.
5. Data sebagai dasar daftar perguruan tinggi, data total mahasiswa, data total tenaga kerja perguruan tinggi berdasarkan data pada PDDIKTI.
6. Data total publikasi akademisi sebagai salah satu kriteria penilaian kuantitatif berdasarkan database *Science and Technology Index* (SINTA) yang mengadopsi database dari Scopus dan Google Scholar.
7. Data akreditasi perguruan tinggi berdasarkan data dari Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT).
8. Metode *LDA-AdaBoost.MH* untuk mengidentifikasi topik dari dokumen abstrak *paper* akademisi perguruan tinggi yang akan menjadi skor nilai reputasi akademik yang bersifat kualitatif.
9. Menggunakan 50 dataset berupa dokumen abstrak *paper* akademisi dari masing-masing perguruan tinggi dengan kategori rumpun teknik.
10. Menggunakan corpus *Taxonomy Bloom* sebagai dasar penetapan corpus kata (*term*) untuk setiap level Tingkat Kesiapterapan Teknologi (TKT).

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini sebagai berikut:

1. Mampu menerapkan teknik *grabbing* terhadap data online berupa informasi *numeric* sebagai masukan kriteria penilaian perguruan tinggi yang bersifat kuantitatif.
2. Mampu menerapkan metode *LDA-AdaBoost.MH* untuk memodelkan topik dari sebuah dokumen abstrak paper akademisi perguruan tinggi sebagai masukan kriteria penilaian yang bersifat kualitatif.
3. Mampu menghasilkan skor dari setiap kriteria penilaian perguruan tinggi, serta skor final penilaian secara keseluruhan/kumulatif dari setiap kriteria.
4. Mampu menganalisa hasil keluaran sistem pemeringkatan dengan *Ground Truth QS World University Rankings*.
5. Mampu menyajikan informasi hasil pemeringkatan perguruan tinggi dalam antarmuka yang jelas dan mudah untuk dimengerti dan diakses oleh pengguna perangkat lunak website (*End User*).

1.5 Manfaat

Tugas Akhir ini diharapkan dapat membantu menghasilkan pemeringkatan perguruan tinggi di Indonesia yang mudah, efektif, dan sederhana berdasarkan Tingkat Kesiapterapan Teknologi (TKT) yang menjadi dasar pemetaan tingkat kematangan teknologi perguruan tinggi di Indonesia. Mengoptimasi kriteria diharapkan mampu menekan waktu kerja (*runnning time*) dan biaya (*cost*) untuk proses pemeringkatan perguruan tinggi.

Penilaian kriteria yang dibahas mengacu pada publikasi pemeringkatan perguruan tinggi internasional tahunan *QS World University Rankings* yang dapat digunakan sebagai tolok ukur kualitas pendidikan tinggi di Indonesia.

1.6 Metodologi

1.6.1 Studi literatur

Pada tahap ini dilakukan studi literatur dengan mencari referensi yang berfokus pada pembahasan teknik *grabbing* data online dari website yang berfungsi sebagai penyedia informasi kriteria penilaian yang bersifat kuantitatif, mencari referensi proses teks mining metode *LDA-AdaBoost.MH* untuk proses topik modelling dokumen abstrak *paper* akademisi perguruan tinggi yang menjadi salah satu kriteria penilaian yang mempunyai bobot ketergantungan paling tinggi, serta mencari referensi literatur lain yang membahas mengenai dasar-dasar dan proses pemerinkatan perguruan tinggi yang sudah ada sebelumnya.

1.6.2 Analisis dan Perancangan Sistem

Pada tahap ini dijabarkan mengenai proses diagram kasus sistem (*use case*) dan aktor yang terlibat dalam alur sistem. Aktor yang menjadi pelaku dalam sistem adalah pengguna perangkat lunak website (*end user*). Kemudian beberapa kebutuhan fungsional dari sistem ini antara lain:

- i. Memasukkan prosentase bobot kriteria penilaian pemerinkatan.
- ii. Melihat peringkat perguruan tinggi berdasarkan tipe pemerinkatan.
- iii. Mencari perguruan tinggi berdasarkan potongan nama perguruan tinggi.
- iv. Mencari perguruan tinggi berdasarkan region wilayah provinsi.

Melihat detail informasi akademik perguruan tinggi.

1.6.3 Implementasi Sistem

Pada tahap ini dilakukan pembuatan perangkat lunak untuk sistem server (*developer*) berupa penerapan teknik *grabbing* pada pengambilan data online, proses teks mining menggunakan metode *LDA-AdaBoost.MH* untuk mengganti kriteria reputasi akademik yang sebelumnya dikerjakan secara manual, serta membangun database dan antarmuka untuk menampilkan informasi akhir kepada pengguna.

1.6.4 Uji Coba dan Evaluasi

Pada tahap ini dilakukan pengujian terhadap perangkat lunak dengan menggunakan dataset final berupa 50 dokumen abstrak *paper* akademisi dari masing-masing perguruan tinggi sebagai pengujian kriteria penilaian yang sifat kualitatif, kemudian menggunakan data online dari database akademisi *Science and Technology Index* (SINTA) sebagai pengujian kriteria penilaian yang bersifat kuantitatif yang mempunyai nilai berupa numeric, serta menggunakan data hasil analisa Backlink, Domain Authority dan Page Authority dari kriteria berbasis analisa *webhost*.

Uji coba dan evaluasi dilakukan untuk mengevaluasi jalannya perangkat lunak, mengevaluasi fitur utama, mengevaluasi fitur tambahan, mencari kesalahan antara hasil peringkat yang dihasilkan oleh sistem perangkat lunak dengan peringkat *Ground Truth* yang sudah ditetapkan, menganalisa faktor-faktor yang mempengaruhi kebenaran sistem, dan mengadakan perbaikan jika ada kekurangan.

1.6.5 Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan pendokumentasian dan pelaporan dari seluruh konsep, dasar teori, implementasi sistem, rancangan antarmuka, proses lain yang telah dilakukan, hasil-hasil dan analisa yang telah didapatkan selama pengerjaan Tugas Akhir.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini memiliki tujuan untuk memberikan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat memberikan manfaat kepada pembaca yang tertarik dalam mengembangkan ke tahap yang lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas 7 bagian dengan isi sebagai berikut.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Dasar Teori

Bab ini membahas teori-teori dasar yang menunjang pokok pembahasan dalam pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak yang meliputi perancangan alur, proses dan perancangan antarmuka pada perangkat lunak.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan perangkat lunak dan implementasi fitur-fitur penunjang perangkat lunak.

Bab V Uji Coba dan Evaluasi

Bab ini membahas pengujian dengan metode pengujian subjektif untuk mengetahui penilaian aspek kegunaan (*usability*) dari perangkat lunak dan pengujian fungsionalitas yang dibuat dengan memperhatikan keluaran yang dihasilkan serta evaluasi terhadap fitur-fitur perangkat lunak.

Bab VI Kesimpulan

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran/rekomendasi untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk pengerjaan Tugas Akhir.

[Halaman ini sengaja dikosongkan]

BAB II

DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir.

2.1 University Rankings

Peringkat perguruan tinggi merupakan salah satu cara untuk mengukur kualitas pendidikan di sebuah negara yang diterapkan oleh seluruh perguruan tinggi di dunia. Namun saat ini tujuan pemeringkatan perguruan tinggi cenderung berbeda dari awalnya yaitu menjadi salah satu cara untuk melihat kepopuleran perguruan tinggi tersebut di mata dunia internasional. Dimana perguruan tinggi yang mempunyai posisi peringkat tinggi akan menjadi perhatian bahwa pendidikan tinggi di negara tersebut berkualitas (K.S. Reddy, En Xie, Qingqing Tang, 2016)- (Tatiana Sidorenko, Tatiana Gorbatova, 2015).

Hingga saat ini, ada berbagai macam publikasi pemeringkatan perguruan tinggi yang populer untuk memberikan hasil pemeringkatan perguruan tinggi di seluruh dunia (Wikipedia, 2016)- (Adina - Petruta Pavel, 2015), diantaranya:

1. QS World University Rankings.
2. Timer Higher Education World University Rankings (THE).
3. Academics Ranking of World University (ARWU).
4. Global Universities Ranging by US News.
5. Reuters World's Top 100 Innovative University.
6. SCImago Institution Rankings.
7. Webometrics.
8. 4ICU (4 International Colleges and Universities).

2.2 QS World University Rankings

Quacquarelli Symonds atau lebih sering disebut dengan *QS World University Ranging* merupakan publikasi peringkat perguruan tinggi seluruh dunia yang berlokasi di Inggris (*QS World University Rankings*, 2016). *QS World University Ranging* diterbitkan setiap tahun dengan menghasilkan perubahan peringkat

yang sangat sensitif, sehingga persaingan antar perguruan tinggi begitu kuat (K.S. Reddy, En Xie, Qingqing Tang, 2016).

Awal mula terbentuknya *QS World University Rankings* pada tahun 2004 merupakan hasil kolaborasi dengan THE (*Times Higher Education*) dengan nama publik THE-QS, namun sejak tahun 2009 QS dan THE sepakat untuk berjalan sendiri-sendiri. QS berjalan hingga sekarang menggunakan metode yang diterapkan sejak awal dan lebih populer disebut publikasi *QS World University Ranking*. Sedangkan THE membangun publikasi peringkat perguruan tinggi sendiri dengan nama THES yang menggunakan metode lain yang hampir serupa dengan *QS World University Ranking* (Mu-Hsuan Huang, 2012).

QS World University Ranking mempunyai 5 cakupan region sebagai observasi pemeringkatan, diantaranya Asia, Amerika Latin, negara Eropa berkembang - Asia Tengah, region Arab dan BRICS (Brasil, Rusia, India, China, South Africa). Untuk menghasilkan peringkat perguruan tinggi, *QS World University Ranking* mempunyai 6 kriteria utama (*QS World University Rankings*, 2016), diantaranya:

1. **Reputasi Akademik (40%)**

Kriteria ini merupakan hasil survey responden akademisi perguruan tinggi di seluruh dunia. Survey ini diharapkan akan mendapatkan hasil yang objektif dari semua responden yang tersebar diseluruh dunia. Mereka mendapatkan bagian survey sesuai dengan latar belakang akademisi tersebut. Terdapat 5 garis besar bidang latar belakang akademisi sebagai responden, yaitu bidang ilmu seni dan sastra, ilmu teknik dan teknologi, ilmu biologi dan kesehatan, ilmu alam, dan ilmu sosial (Mu-Hsuan Huang, 2012).

2. **Reputasi Employer (10%)**

Kriteria ini merupakan hasil survey responden perusahaan yang menaungi alumni perguruan tinggi di seluruh dunia (Mu-Hsuan Huang, 2012). Sama dengan kriteria reputasi akademik, kriteria ini ditujukan agar memperoleh hasil yang objektif terhadap kualitas lulusan (fresh graduate) perguruan

tinggi. Salah satu faktor penting agar perguruan tinggi dapat diperhatikan dalam kriteria ini adalah dengan menjalin banyak kerjasama dengan perusahaan sebagai rekan kerja maupun penelitian (Tatiana Sidorenko, Tatiana Gorbatoва, 2015).

3. Jumlah Mahasiswa Aktif (20%)

Kriteria ini merupakan hasil jumlah mahasiswa yang sedang menempuh pendidikan di perguruan tinggi. Tujuan dari kriteria ini adalah untuk menganalisa pengawasan jumlah infrastruktur dengan jumlah mahasiswa, agar jumlah infrastruktur sebanding dengan jumlah mahasiswa. Perguruan tinggi yang baik merupakan perguruan tinggi yang memberikan fasilitas dan kenyamanan belajar (Mu-Hsuan Huang, 2012).

4. Jumlah Sitasi Akademisi (20%)

Kriteria ini merupakan hasil jumlah sitasi maupun publikasi dari akademisi perguruan tinggi yang *ter-index* oleh publikasi Scopus. Sitasi/publikasi yang dihitung berusia maksimal terbit 5 tahun sebelumnya. Tujuan dari kriteria ini adalah untuk melihat seberapa banyak dampak penelitian akademisi perguruan tinggi terhadap lingkungan (Mu-Hsuan Huang, 2012). Karena tujuan perguruan tinggi adalah yang dapat memberikan dampak baru bagi masyarakat.

5. Jumlah Tenaga Pengajar Internasional (5%)

Kriteria ini merupakan hasil dari jumlah tenaga pengajar (staff/dosen) asing di sebuah perguruan tinggi. Lama waktu minimal tenaga pengajar asing menetap selama 3 bulan per tahun (Tatiana Sidorenko, Tatiana Gorbatoва, 2015). Tujuan dari kriteria ini adalah melihat kerjasama dan ketertarikan tenaga asing untuk perguruan tinggi tersebut.

6. Jumlah Mahasiswa Asing (5%)

Kriteria ini merupakan hasil dari jumlah mahasiswa asing yang belajar di perguruan tinggi, baik dalam rangka pertukaran pelajar, maupun program pendidikan lainnya (Mu-Hsuan Huang, 2012). Tujuan dari kriteria ini adalah untuk melihat ketertarikan mahasiswa asing terhadap metode pembelajaran perguruan tinggi (Tatiana Sidorenko, Tatiana Gorbatova, 2015).

Dalam penilaian kriteria *QS World University Ranking* terdapat 2 jenis kriteria, yaitu kriteria yang bersifat kualitatif dan kuantitatif. Untuk kriteria yang bersifat kualitatif merupakan kriteria yang membutuhkan pengolahan data untuk mendapatkan nilai serta cenderung bersifat deskriptif dan membutuhkan analisa terlebih dahulu. Kriteria yang termasuk dalam jenis kualitatif adalah kriteria reputasi akademik dan reputasi employer.

Sedangkan untuk kriteria yang bersifat kuantitatif merupakan kriteria yang dapat ditentukan nilainya secara langsung tanpa melakukan analisa terlebih dahulu dan tidak bersifat deskriptif. Kriteria yang termasuk dalam jenis kuantitatif adalah kriteria jumlah mahasiswa aktif, jumlah sitasi/publikasi akademisi, jumlah staff internasional dan jumlah mahasiswa asing.

Untuk kriteria pertama yang bersifat kualitatif, yaitu reputasi akademik dalam *QS World University Rankings* ada beberapa poin yang menjadi acuan untuk digali dari responden survey untuk menghasilkan penilaian yang objektif (QS World University Rankings, 2014), diantaranya:

1. Biodata personal responden akademisi

Poin penilaian ini, responden diminta menjelaskan mengenai biodata personal responden yang berkaitan dengan dimana akademisi bekerja, diantaranya seperti nama lengkap, alamat email institusional, jobdesk pekerjaan, lama bekerja dalam institusi, nama institusi dimana akademisi bekerja, pendapat mengenai negara yang paling berpengaruh dalam pendidikan tinggi dan penelitian, serta menanyakan mengenai pengetahuan akademisi terhadap Massive Open

Online Courses (MOOCs) (QS World University Rankings, 2014).

2. Knowledge specification

Poin penilaian ini, responden diminta menjelaskan berdasar pengetahuan responden mengenai pengetahuan dasar tentang region dan bidang ilmu yang akan dibahas/ditanyakan, diantaranya seperti pengetahuan terhadap kualitas akademik berdasarkan region (amerika, asia-australia-new zealand, eropa-timur tengah-afrika atau semua region), pengetahuan terhadap kategori bidang ilmu pengajaran, pengetahuan terhadap sub bidang ilmu yang lebih spesifik, serta mengenai pembagian waktu akademisi untuk melakukan riset (*research*), pengajaran (*teaching*), administrasi (*administration*) dan lainnya dalam bentuk persen (QS World University Rankings, 2014).

3. Penelitian terbaik

Poin penilaian ini, responden diminta memilih beberapa perguruan tinggi yang dianggap mempunyai tingkat penelitian (*research*) yang baik, diantaranya seperti memilih 10 institusi dalam negeri sesuai dengan region negara responden yang mempunyai kualitas penelitian terbaik, memilih 30 institusi luar negeri yang mempunyai kualitas penelitian terbaik, serta dalam memilih institusi dalam negeri maupun luar negeri tersebut harus mencakup kategori bidang ilmu yang diteliti. Ada 5 bidang ilmu, maka total responden harus memilih 50 perguruan tinggi dalam negeri terbaik yang mencakup 5 bidang ilmu, serta 150 perguruan luar negeri yang mencakup 5 bidang ilmu.

4. Inovasi dalam penelitian

Poin penilaian ini, responden diminta mengidentifikasi perguruan tinggi mana yang mempunyai tingkat inovasi

yang baik dalam penelitian, diantaranya seperti mengidentifikasi institusi yang paling cepat dan dinamis dalam perkembangan dan inovasi penelitian, serta bidang ilmu apa yang dimiliki institusi tersebut yang paling cepat berkembang dalam hal inovasi (QS World University Rankings, 2014).

5. Leading company

Poin penilaian ini, responden diminta mengidentifikasi perusahaan mana yang paling berkontribusi terhadap penelitian, diantaranya seperti mengidentifikasi 5 perusahaan yang menurut responden akademisi paling berpengaruh terhadap pendidikan sesuai dengan bidang pendidikan responden, serta menjelaskan alasan memilih 5 perusahaan tersebut (QS World University Rankings, 2014).

6. MOOCs

Poin penilaian ini, responden diminta menjelaskan mengenai penerapan MOOCs dalam institusi responden (Vidya Rajiv Yeravdekar, Gauri Tiwari, 2014), diantaranya seperti apakah institusi pendidikan responden sudah menerapkan pembelajaran berbasis MOOCs, apakah institusi pendidikan responden dapat merencanakan pembelajaran berbasis MOOCs untuk 3 tahun kedepan, apakah responden telah menerapkan pembelajaran MOOCs berdasarkan inisiatif pribadi, serta pendapat responden mengenai keuntungan pembelajaran berbasis MOOCs (QS World University Rankings, 2014).

7. Profil penelitian responden akademisi

Poin penilaian ini, responden diminta menjelaskan profil publikasi/penelitian responden selama ini jika ada, diantaranya seperti artikel jurnal, konferensi, paper/buku, kemampuan (*performance*) responden, pengalaman (*exhibitions*) responden, serta paten yang dimiliki (QS World University Rankings, 2014).

Untuk kriteria kedua yang bersifat kualitatif yaitu reputasi employer, ada beberapa poin yang menjadi acuan untuk digali dari responden perusahaan untuk menghasilkan penilaian yang objektif (QS World University Rankings, 2014), diantaranya:

1. Biodata personal responden perusahaan

Poin penilaian ini, responden perusahaan diminta menjelaskan mengenai biodata personal yang berkaitan dengan institusi dimana responden bekerja, diantaranya nama lengkap, nama perusahaan, alamat email resmi berdomain perusahaan, website perusahaan, jabatan pekerjaan, negara lokasi perusahaan berjalan, sektor utama perusahaan, ukuran perusahaan (kuantitas staff), kategori lulusan yang diterima sebagai staff (internship, sarjana-magister (S1/S2), MBA), dan yang terakhir lokasi negara yang paling banyak kandidat diterima sebagai staff perusahaan (QS World University Rankings, 2014).

2. Analisa perguruan tinggi bisnis terbaik

Poin penilaian ini, responden perusahaan diminta menjelaskan mengenai pendapat mereka terhadap perguruan tinggi yang mempunyai lulusan terbaik, baik dalam hal bisnis maupun keseluruhan, diantaranya adalah memilih 10 perguruan tinggi domestik sesuai tempat tinggal responden yang mempunyai lulusan terbaik, memilih 30 perguruan tinggi luar negeri yang mempunyai lulusan terbaik, memberi tahu daftar perguruan tinggi dari staff perusahaan yang bekerja dalam perusahaan tersebut, serta bidang disiplin ilmu yang paling banyak direkrut oleh perusahaan tersebut (QS World University Rankings, 2014).

Selain itu mengidentifikasi juga sekolah bisnis dimana responden perusahaan diminta untuk menjelaskan beberapa hal, diantaranya adalah mengidentifikasi 10 sekolah bisnis yang alumennya berpotensi direkrut oleh perusahaan, perekrutan staff bergelar MBA (Master of Business Administration), membiayai staff pada program executive MBA untuk mengembangkan potensi,

menjelaskan pentingnya spesialisasi MBA dalam perusahaan, serta memilih 5 perguruan tinggi yang unggul dalam gelar MBA (ada 10 kategori spesialisasi, yang berarti ada 50 perguruan tinggi yang dipilih).

3. Kualifikasi untuk staff bergelar MBA

Poin penilaian ini, responden perusahaan diminta menjelaskan pendapatnya mengenai kualifikasi staff yang bergelar MBA, diantaranya adalah memilih region negara yang mempunyai sekolah bisnis terbaik dan menciptakan dampak tinggi untuk bisnis, lalu menjelaskan mengenai ketentuan yang ideal untuk staff bergelar MBA untuk bekerja (lama waktu kerja per hari, gaji, bonus, tunjangan), divisi/posisi kerja yang sesuai dengan staff bergelar MBA, rata-rata gaji yang ditawarkan oleh perusahaan untuk freshgraduate.

4. Pembahasan umum

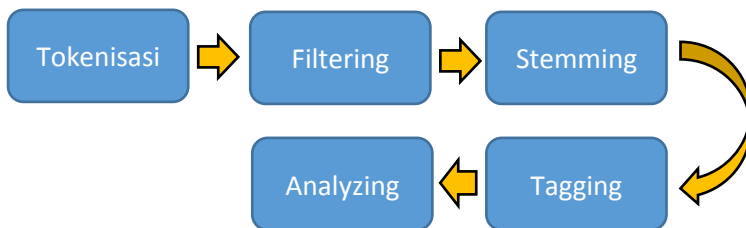
Poin penilaian ini, responden perusahaan diminta menjelaskan pendapatnya mengenai kebutuhan skill apa yang harus dimiliki oleh calon staff (bergelar MBA maupun reguler), diantaranya adalah skill yang dibutuhkan untuk freshgraduate, seperti pengalaman internasional, pengetahuan tentang subjek, kepemimpinan, pengalaman magang, kemampuan individu, dan kemampuan bahasa. Selain itu diminta menjelaskan mengenai skill yang dibutuhkan oleh calon staff MBA, seperti kemampuan individu, interaksi dengan lingkungan, kepemimpinan, prestasi akademik, pemikiran strategis, kemampuan bahasa, kewiraswastaan, keterampilan IT, manajemen resiko, pengalaman, pemasaran, keterampilan keuangan, komunikasi, tanggung jawab sosial, dan e-business (QS World University Rankings, 2014).

Untuk kriteria ketiga hingga keenam, poin yang dijadikan penilaian adalah jumlah dari masing-masing. Kriteria ini tidak mempunyai

ketentuan poin khusus seperti kriteria sebelumnya yang bersifat kualitatif.

2.3 Teks Mining

Teks mining merupakan proses menggali data teks yang didapatkan dari sumber data berupa dokumen (word, pdf, kutipan, dan sejenisnya). Teks mining disebut juga dengan teknik mengekstraksi pola dari sebuah dokumen. Tujuan utama dari teks mining adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen dengan melakukan pencarian kata-kata yang dapat mewakili isi dari sebuah dokumen yang kemudian dapat dianalisa. Kegunaan dari penggunaan teks mining adalah sebagai pengkategorian teks (*text categorization*) dan pengelompokan teks (*text clustering*) (Raymond J. Mooney, 2000). Proses umum dalam teks mining seperti pada Gambar 2.3-1.



Gambar 2.3-1 Proses Utama Teks Mining

2.4 Teknik Grabbing

Merupakan salah satu cara untuk mengambil konten teks atau object yang ditampilkan dalam situs website yang kemudian akan diletakkan dalam website milik pribadi. Secara umum teknik ini akan menduplikasi konten website tujuan untuk ditampilkan dalam website kita, akan tetapi dari sini banyak informasi dan kegunaan lain yang dapat dikembangkan. Salah satunya adalah sebagai salah satu bahan teks mining dimana dengan tujuan untuk menganalisa konten teks dalam website tersebut. Teknik grabbing lebih banyak dikembangkan menggunakan bahasa pemrograman

PHP, terdapat library yang mendukung analisa ini. Teknik grabbing dapat dikembangkan sendiri oleh developer sesuai dengan kebutuhan yaitu kegunaan dan konten situs website sasaran, oleh karena itu teknik ini bisa dikatakan tidak fleksibel karena jika ingin melakukan grabbing pada situs website tertentu maka harus merubah pola konten yang akan diambil. Beberapa cara menggunakan teknik grabbing diantaranya adalah fungsi cURL, fungsi file_get_contents, library simple HTML dom parser, dan pemanfaatan ReGex (*Regular Expression*).

2.4.1 Fungsi cURL

Merupakan sebuah fungsi yang berguna untuk mengambil konten dari sebuah situs website. cURL dapat digunakan untuk melakukan simulasi request method POST ke server dan juga menggunakan COOKIE. Fungsi ini dijalankan pada Bahasa pemrograman PHP. Keuntungan menggunakan cURL adalah beberapa situs website yang melindungi kontennya dengan menggunakan cookie dan post method, dengan menggunakan cURL dapat mengakses konten yang terproteksi tersebut. Beberapa fungsi yang diakomodasi oleh cURL:

1. curl_init(), untuk inisialisasi curl library.
2. curl_setopt(), untuk mengatur pilihan dari curl.
3. curl_exec(), untuk mengeksekusi/menjalankan query.
4. curl_close(), untuk menghentikan curl system.

```
$options = array(
    CURLOPT_CUSTOMREQUEST => "GET",
    CURLOPT_POST           => false,
    CURLOPT_FOLLOWLOCATION => true,
    CURLOPT_CONNECTTIMEOUT => 120,
    CURLOPT_TIMEOUT       => 120
);

$ch = curl_init( $url );
curl_setopt_array( $ch, $options );
$content = curl_exec( $ch );
curl_close( $ch );
$header['content'] = $content;
```

Kode Sumber 2.4-1 Syntax cURL dengan setting option url

2.4.2 Fungsi File_get_contents

Merupakan cara untuk membaca isi file ke dalam string. Karena akan menggunakan teknik pemetaan memori, jika hal ini didukung oleh server, untuk meningkatkan kinerja. Isi file dalam konteks grabbing adalah, bagian konten situs website yang akan diterjemahkan menjadi string, kemudian akan dianalisa sesuai kebutuhan dengan mengambil informasi dari tag konten bagian tertentu menggunakan ReGex (*Regular Expression*).

```
file_get_contents(path,include_path,context,start,max_length)
```

Kode Sumber 2.4-2 Syntax Fungsi file_get_contents

2.4.3 Simple HTML Dom Parser

Merupakan sebuah syntax dalam Bahasa pemrograman PHP yang berfungsi untuk menguraikan elemen-elemen HTML untuk diambil dan dirubah atau dihapus pada tag/elemen-elemen tertentu menjadi konten baru yang diinginkan. Dalam PHP sendiri sudah disediakan library HTML DOM yang dapat diunduh terpisah dengan banyak keuntungan.

```
<?php
    require_once 'simple_html_dom.php';

    $html = file_get_html("http://hieppies.blogspot.com");
    $html = file_get_html("index.html");
    $html = file_get_html("/menu/index.html");

    $html = str_get_html("<html>
        <body>
            <p>Hi, Cantik!.</p>
            <p>Lagi ngapain?.</p>
        </body>
    </html>");

?>
```

Kode Sumber 2.4-3 Syntax Penggunaan Library Simple HTML Dom Parser

2.4.4 Regular Expression (ReGex)

Merupakan teknik yang digunakan untuk pencocokan string teks, seperti karakter tertentu, kata-kata, atau pola karakter. ReGex memiliki 2 fungsi utama yaitu untuk mencari dan mengganti, mencari suatu pola tertentu dalam text lalu menggantinya menjadi pola yang lain. Dalam penggunaannya untuk grabbing situs website, ReGex biasanya dikombinasikan dengan fungsi `file_get_contents` dari PHP. Mengambil konten menggunakan fungsi `file_get_contents` sedangkan menganalisa hasil dan mencari string konten yang dibutuhkan menggunakan regular expression.

2.5 Pusat Data *Scopus*

Scopus merupakan database bibliografi yang meng-index abstrak dan sitasi/kutipan untuk artikel jurnal akademik. Scopus merupakan salah satu produk bibliograsi dari Elsevier. Scopus muncul sejak tahun 1995, kurang lebih 21 tahun Scopus meng-index jurnal dari para akademisi di seluruh dunia. Hingga saat ini kurang lebih terdapat 22.000 judul artikel jurnal dan lebih dari 5.000 penerbit (Wikipedia, 2016).

Scopus meng-index artikel jurnal akademik dari berbagai bidang ilmu. Maka dari itu banyak publikasi pemeringkatan perguruan tinggi lain seperti Scimago menggunakan jasa dari Scopus sebagai pusat database jurnal akademik. Scopus sendiri sangat berhati-hati dan selektif untuk meng-index artikel jurnal akademik. Tidak semua artikel jurnal di database lain seperti Google Scholar, ataupun Web of Science dapat ter-index oleh Scopus. Scopus hanya meng-index berdasarkan kualitas artikel jurnal tersebut.

Scopus juga menyediakan layanan untuk developer berupa Scopus API (Application Program Interfaces) sebagai sarana untuk mengembangkan informasi berdasarkan database Scopus, hanya saja untuk menggunakan layanan ini hanya untuk institusi yang berlangganan (*Subscribe*) saja (Elsevier, 2016). Tidak semua orang dan institusi dapat menggunakan layanan ini. Keuntungan dari

database Scopus ini adalah mempunyai atribut data dan informasi yang lengkap, diantaranya nama author, jumlah publikasi, jumlah sitasi, ORCID, dan grafik perkembangan keaktifan author (Elsevier, 2016). Selain itu dukungan layanan API sangat lengkap dan juga kontennya dapat dipercaya.

2.6 *Google Scholar*

Google Scholar merupakan layanan database metadata publikasi yang dibuat oleh Google. Dalam Google Scholar memungkinkan pengguna melakukan pencarian materi berupa teks dalam berbagai format publikasi. Index Google Scholar mencakup jurnal online dan publikasi ilmiah sehingga semua jurnal online bisa tercover dalam Google Scholar (Anne-Wil K. Harzing, Ron van der Wal, 2008). Tujuan utama dari Google Scholar adalah menyusun paper seperti yang dilakukan peneliti, dengan memperhatikan kelengkapan teks, nama penulis yang menampilkan artikel, serta frekuensi penggunaan kutipan artikel dalam literatur akademis lainnya.

Dalam Google Scholar, seorang peneliti dapat mendaftarkan akun dirinya dalam sistem sebagai biodata untuk setiap paper yang dipublikasikan. Dalam setiap biodata penulis terdapat beberapa indikator yang cukup unik sebagai bahan acuan bahwa paper yang dipublikasikan oleh penulis tersebut berkualitas dan terpercaya atau tidak (Wikipedia, 2017). Hal ini dapat dilihat dari jumlah paper yang dipublikasikan penulis yang terindex oleh Google Scholar, jumlah sitasi (*citation*), nilai h-index, dan nilai i10-index (Wikipedia, 2017). Berikut penjelasan masing-masing indikator dalam Google Scholar:

1. Jumlah Publikasi

Merupakan jumlah paper yang dipublikasikan oleh penulis yang terindex oleh Google Scholar. Disini dapat diketahui juga tahun publikasi dan jumlah sitasi/kutipan terhadap paper tersebut. Semakin banyak orang lain yang mengutip paper tersebut akan semakin tinggi peringkat posisi paper dalam daftar publikasi penulis tersebut. Jumlah publikasi ini dihitung sejak penulis aktif dalam Google Scholar.

2. Jumlah Sitasi

Merupakan jumlah total kutipan dari orang lain yang mengutip paper penulis tersebut. Semakin banyak orang yang mengutip paper penulis maka akan semakin banyak pula total sitasinya. Terdapat dua versi dalam jumlah sitasi pada Google Scholar, terdapat jumlah total (*all citation*) yang merupakan jumlah keseluruhan sejak penulis aktif dalam Google scholar, serta jumlah sejak 5 tahun terakhir (Since 2012) yang dapat dilihat bahwa kualitas sitasi dari paper tersebut baik atau tidak sejak 5 tahun terakhir.

3. Nilai H-Index

Merupakan ukuran index yang diusulkan oleh seorang fisikawan di Universitas California bernama Jorge E Hirsch pada tahun 1985. H-Index ini masih terus diteliti keakuratannya. H-Index sendiri adalah nilai index yang didapatkan dari jumlah paper sebanyak h paper penulis dengan jumlah kutipan untuk setiap *paper* minimal sama dengan nilai h tersebut (Wikipedia, 2017).

4. Nilai i10-Index

Merupakan nilai index yang didapatkan dari jumlah 2 *paper* dari penulis yang dikutip minimal oleh 10 *paper* dari orang lain. Sehingga kondisi minimal dari kualitas *paper* tersebut dapat diketahui oleh cara ini.

2.7 *Science and Technology Index (SINTA)*

Science and Tecnology Index atau biasa disebut dengan SINTA merupakan pusat index, sitasi dan kepakaran terbesar di Indonesia. Sistem informasi riset berbasis website ini menawarkan akses cepat, mudah dan komprehensif untuk mengukur unjuk kerja peneliti, institusi dan jurnal di Indonesia. Sinta menyediakan benchmark dan analisis, identifikasi kekuatan riset tiap institusi, memperlihatkan kolaborasi penelitianm menganalisis tren penelitian dan direktori pakar. SINTA memberikan informasi mengenai tolok ukur institusi, kolaborasi, analisis penelitian serta direktori akademisi di Indonesia (Kementerian Riset, Teknologi

dan Pendidikan Tinggi, 2017). Beberapa informasi yang dapat digali dari database online SINTA (Kementerian Riset, teknologi dan Pendidikan Tinggi, 2017), antara lain:

1. Jurnal yang diterbitkan oleh institusi pendidikan dan penelitian tinggi yang disajikan dalam bentuk peringkat.
2. Institusi pendidikan tinggi yang mempunyai kontribusi dalam penelitian, berisikan informasi jumlah author akademisi serta total sitasi yang dihasilkan dari masing-masing institusi pendidikan dan penelitian tinggi yang disajikan dalam bentuk peringkat.
3. Author akademisi yang mempunyai kontribusi dalam penelitian, berisikan informasi jumlah sitasi yang dimiliki, skor h-index, dan i10-index.
4. Detail informasi setiap institusi pendidikan tinggi:
 - a. Jumlah sitasi Google Scholar.
 - b. Jumlah dokumen Google Scholar.
 - c. Jumlah artikel pada Scopus database.
 - d. Jumlah non artikel pada Scopus database.
 - e. Jumlah sitasi dari Scopus.
 - f. Jumlah akademisi.
 - g. Daftar akademisi beserta kualitas sitasi (h-index, i10-index) dari Google Scholar dan Scopus database.
 - h. Jumlah jurnal artikel, book chapters, dan conference paper dari akademisi maupun institusi.
 - i. Skor penilaian menurut SINTA.

2.8 Pangkalan Data Pendidikan Tinggi (PDDIKTI)

Pangkalan Data Pendidikan Tinggi atau lebih umum disebut PDDIKTI merupakan pusat kumpulan data penyelenggara pendidikan tinggi seluruh Indonesia. Kementerian Riset, Teknologi, dan Pendidikan Tinggi merupakan kementerian dalam pemerintahan Indonesia yang berfokus pada penyelenggaraan urusan bidang riset, teknologi dan pendidikan tinggi (Kementerian Riset Teknologi dan Pendidikan Tinggi Republik Indonesia, 2016).

Data dalam PDDIKTI merupakan hasil sinkronisasi yang dikelola oleh masing-masing perguruan tinggi di Indonesia, sehingga akan lebih update dalam penyebaran informasi, serta terintegrasi secara nasional. Pangkalan data ini menjadi salah satu instrumen pelaksanaan penjaminan mutu. Data yang terangkum dalam PDDIKTI harus mudah diakses oleh stakeholder (lembaga akreditasi, pemerintah, masyarakat), sehingga menerapkan prinsip transparansi perguruan tinggi.

Beberapa data yang disediakan oleh PDDIKTI dalam sistem yang dapat dilihat (Kementerian Riset Teknologi dan Pendidikan Tinggi Republik Indonesia, 2016), diantaranya:

1. Jumlah perguruan tinggi di Indonesia.
2. Jumlah program studi setiap perguruan tinggi.
3. Jumlah mahasiswa (jenis kelamin, kelompok bidang).
4. Jumlah dosen (jenis kelamin, ikatan kerja/status kepegawaian, pendidikan terakhir, jabatan akademik).

2.9 Alexa Traffic Rank

Alexa Traffic Rank merupakan sistem yang menyediakan fasilitas informasi tentang peringkat *webhost*, yang dilihat dari jumlah *traffic* pengunjung yang masuk kedalam situs *webhost* tersebut. Dengan kata lain Alexa Traffic Rank merupakan salah satu *tools* untuk *Search Engine Optimization* (SEO) website yaitu sitem untuk menganalisa dan mempengaruhi tingkat keterlihatan (*visibilitas*) sebuah situs pada mesin pencari (Arjun Thakur, A. L. Sangal, Harminder Bindra, 2011).

Alexa Traffic Rank mempunyai nilai peringkat dari nilai terkecil yaitu nilai angka 1 (satu) hingga seterusnya (berjuta-juta) yang mana dalam perhitungan sistem di Alexa Traffic Rank apabila semakin kecil nilai Alexa Traffic Rank, nilai kepopuleran situs akan semakin bagus. Selain itu Alexa Traffic Rank juga menyediakan API untuk pengembangan (*developer*) perangkat lunak berbasis analisa SEO yang tidak berbayar (*free*).

2.10 MOZ Analytics

MOZ Analytics merupakan salah satu *tools* yang digunakan untuk mengetahui jumlah pencarian *webhost*. MOZ Analytics dikembangkan oleh SEOMoz yang berbasis di Seattle, Washington, Amerika Serikat. Salah satu product dari MOZ yang berfokus pada pemeringkatan dan analisa adalah MOZ Rank. MOZ Rank mengkuantifikasi kepopuleran link *webhost*. Halaman *webhost* mendapatkan skor MOZ Rank berdasarkan halaman lain yang terhubung dengan halaman *webhost* tersebut (ditautkan). Semakin banyak tautan dari halaman *webhost*, maka akan semakin tinggi skor dari MOZ Rank. Dengan kata lain MOZ Analytics dengan berbagai macam produk turunannya merupakan sebuah *tools* untuk *Search Engine Optimization* (SEO). MOZ Analytics mengembangkan API untuk *developer* SEO yang tidak berbayar (*free*).

2.11 Bahasa Pemrograman Python

Python merupakan bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python sendiri mulai diluncurkan di masyarakat sejak tahun 1991 oleh Guido van Rossum dengan nama vendor Python Software Foundation. Python diklaim sebagai bahasa pemrograman yang menggabungkan kapabilitas, kemampuan dengan sintaksis kode yang jelas dan dilengkapi dengan pustaka (*library*) yang lengkap dan komprehensif.

Python didistribusikan dengan beberapa lisensi yang berbeda dari beberapa versi yang diluncurkan. Namun pada prinsipnya python dapat diperoleh dan dipergunakan secara bebas, bahkan untuk kepentingan komersial. Lisensi Python tidak bertentangan dengan definisi Open Source dan General Public License (Wikipedia, 2017). Python terdiri dari dua buah versi terbaru yaitu 3.4.2, dan 2.7.8 yang bebas digunakan oleh developer. Versi Python ini rilis pada tahun 2014. Kelebihan yang dimiliki oleh bahasa pemrograman Python sebagai berikut:

1. Memiliki kepustakaan (*library*) yang luas, dalam Python sudah disediakan modul-modul siap pakai sesuai kebutuhan developer.
2. Memiliki tata bahasa yang mudah dipelajari.
3. Memiliki aturan layout (*indentation*) yang mudah untuk dilakukan pengecekan.
4. Berfokus pada orientasi objek.
5. Memiliki sistem pengolahan memori otomatis.
6. Modular, mudah dikembangkan dengan menciptakan modul-modul baru sesuai kebutuhan.

2.12 Library Python

Beberapa pustaka (*library*) yang digunakan dalam membangun sistem Tugas Akhir yang berhubungan dengan *teks mining*, diantaranya:

1. Natural Language ToolKit (NLTK)

NLTK merupakan pustaka utama untuk proses pengolahan bahasa alami (*natural language process*). Pustaka ini berperan penting dalam proses teks mining seperti klasifikasi, tagging, tokenisasi, dan pemrosesan semantic. Pustaka ini diimplementasikan pada bahasa pemrograman python, selain itu pustaka ini juga didukung oleh dokumentasi yang lengkap, mudah untuk dipahami, dan forum diskusi yang banyak membantu (Steven Bird, 2006). Beberapa pustaka NLTK yang diterapkan dalam Tugas Akhir ini, antara lain:

- a. NLTK corpus stopwords

Merupakan pustaka yang menyediakan corpus berisi keyword kata-kata stopwords, kata umum yang kurang penting keberadaannya didalam konteks dokumen. Corpus stopwords ini dikembangkan dalam berbagai bahasa yang disediakan oleh NLTK.

- b. NLTK Tokenize

Merupakan pustaka untuk melakukan preprocessing dokumen dengan melakukan tokenize dokumen menjadi sebuah kata tunggal.

- c. Porter Stemmer
Merupakan pustaka untuk proses stemming dalam sebuah dokumen, dengan menghapus imbuhan kata. Stemmer ini mengadopsi berbagai tipe bahasa utam yang ada didunia.
- d. Snowball Stemmer
Merupakan pustaka lain untuk proses stemming. Pustaka stemming yang dimiliki NLTK bervariasi dengan tujuan pengembangan untuk mencapai hasil yang lebih bagus. Dari hasil penelitian sebelumnya, proses stemming dalam natural language process (NLP) menjadi isu utama yang harus terus dipecahkan. Snowball Stemmer ini merupakan salah satu pengembangan dari Porter Stemmer yang mempunyai keuntungan lebih.
- e. WordNet
WordNet merupakan database leksikon untuk bahasa Inggris. Fungsi utama pustaka ini adalah untuk mengelompokkan sebuah kata kepada kata lain yang terhubung dalam rule sinonim atau biasa disebut synsets (Tingting Wei, Yonghe Lu, Houyou Chang, Qiang Zhou, Xianyu Bao, 2015). Penggunaan utama dari WordNet sebagai kombinasi antara kamus dan thesaurus.
- f. Postag Word Tokenize
Merupakan pustaka untuk pemrosesan tagging *Part of Speech* (POS Tagger) kata dalam sebuah kalimat dalam dokumen. Pustaka ini berperan penting dalam penerapan perangkat lunak pada Tugas Akhir ini.

2. Sckit Learn

Merupakan pustaka dalam bahasa python yang berfokus pada proses pembelajaran numeric, seperti algoritma klasifikasi, clustering, dan regresi. Pustaka ini dilengkapi dengan berbagai macam algoritma dan didesain berjalan dalam sistem vector Numpy dan SciPy (Fabian Pedregosa,

Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, 2011).

Beberapa algoritma dari pustaka Sckit Learn yang digunakan dalam Tugas Akhir ini, antara lain:

a. CountVectorizer

Pustaka ini ada pada modul `sklearn.feature_extraction` yang digunakan untuk mengekstrak fitur dalam format yang didukung oleh algoritma pembelajaran dari Sckit Learn dari kumpulan data berformat teks maupun gambar. Fungsi utama CountVectorizer adalah mengkonversi kumpulan dokumen teks kedalam matriks token (*count term*).

b. TfidfTransformer

Pustakan ini berfungsi sebagai transformasi matriks perhitungan menjadi representasi TF-IDF yang dinormalisasi. Merupakan skema pembobotan untuk mengurangi term yang sering muncul di corpus tertentu.

3. Numpy

Merupakan modul yang menyediakan objek perhitungan mathematic yang memudahkan dalam melakukan perhitungan. Objek utama yang disediakan adalah array yang dapat berperan sebagai matriks.

4. TextBloob

Textblob merupakan pustakan yang dapat melakukan berbagai proses terhadap data teks, seperti tokenisasi, analisa sentiment hingga proses translate bahasa kedalam beberapa bahasa yang umum didunia. Penggunaan textbloob dalam Tugas Akhir ini sebagai pustaka untuk melakukan uji coba proses translate dokumen berbahasa Indonesia menjadi bahasa Inggris sebelum dilakukan preprocessing (TextBlob, 2017).

5. Option Parser

Merupakan pustaka yang fleksibel untuk mengurai opsi dalam baris perintah pada command line terminal python. Keuntungan dari Option Parser adalah menggunakan gaya deklaratif bahasa yang lebih baik daripada modul yang sama sebelumnya. Option Parser atau Optparse memungkinkan user untuk menentukan opsi dalam sintaks GNU/POSIX dengan didukung oleh pilihan bantuan (*help*).

6. MySQL Python dan MySQL Client

Merupakan modul dari MySQL untuk konektor bahasa pemrograman python kepada basis data MySQL. MySQL Python dan MySQL Client digunakan untuk menyimpan hasil akhir teks mining dalam database.

7. Pustaka import tambahan

Selain menggunakan pustaka utama diatas yang membutuhkan konfigurasi lebih lanjut, ada beberapa pustaka default dari python, diantaranya:

a. String

Modul ini berisi beberapa konstanta dan kelas yang berfungsi sebagai metode pengolah string untuk tipe data string dalam pengolahan kata, seperti fungsi str, unicode, list, tuple, bytearray, buffer, xrange section.

b. Re (regular expression)

Modul ini menyediakan operasi pencocokan regular expression untuk pola tipe data string. Modul ini sesuai untuk pengolahan kata yang tanpa menggunakan library tambahan.

c. Os (Operating System)

Modul ini menyediakan cara portabel untuk menggunakan fungsi yang bergantung pada sistem operasi. Menjadi jalan untuk proses programming untuk memodifikasi fungsi pada sistem operasi.

d. Math

Modul ini menyediakan akses untuk mengeksplere fungsi matematika yang didefinisikan oleh standar

bahasa C. Modul ini tidak dapat digunakan dengan bilangan kompleks.

- e. Random
Modul ini mengimplementasikan proses generator bilangan acak untuk berbagai macam distribusi. Generator bilangan acak dapat ditentukan sesuai kebutuhan pengguna.
- f. Time
Modul ini menyediakan berbagai fungsi yang berhubungan dengan waktu. Meskipun modul ini selalu tersedia, tidak semua fungsi tersedia di semua platform. Sebagian besar fungsi yang didefinisikan dalam modul ini memanggil fungsi library bahasa C.
- g. Csv
Format CSV (Comma Separated Values) merupakan format impor dan ekspor paling umum untuk spreadsheet dan database. Penggunaannya yang sederhana dan kurangnya standar untuk norma penulisan menjadi perbedaan disetiap platform yang menggunakan CSV.

2.13 PHP dan HTML

Hypertext Preprocessor atau yang biasa disebut dengan PHP merupakan bahasa script yang dapat disisipkan dalam kumpulan kode HTML (Wikipedia, 2017). PHP banyak dipakai dalam pemrograman website selain dengan HTML. PHP sendiri ditemukan oleh Rasmus Lerdorf pada tahun 1995, dan hingga sekarang PHP dikembangkan oleh vendor utama yaitu The PHP Group. Hingga saat ini PHP telah masuk dalam versi ke 7, atau biasa yang disebut dengan PHP 7. Kelebihan dari bahasa pemrograman PHP sebagai berikut:

1. Bahasa pemrograman yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web server yang mendukung PHP dapat ditemukan dimana-mana, diantaranya apache, IIS, Lighttpd, dan Xitami.

3. Dalam sisi pengembangan (*developer*) lebih mudah karena banyak organisasi maupun developer lain yang siap membantu mengembangkan variasi dari PHP sendiri.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang baik dan banyak.
5. Open Source yang dapat digunakan di berbagai *Operating System* (OS) dan dapat dijalankan secara *runtime* melalui *console* serta dapat menjalankan perintah sistem.

2.14 Database Management System (DBMS) MySQL

MySQL merupakan sebuah perangkat lunak sistem manajemen basis (DBMS) data SQL. MySQL disebut database yang multithread, multi user dengan jumlah instalasi 6 juta kali di seluruh dunia yang membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL). MySQL sendiri dikembangkan oleh Oracle pada tahun 1995. MySQL sendiri bisa dijalankan disemua tipe sistem operasi (Wikipedia, 2017).

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional RDBMS yaitu sistem basis data yang entitas utamanya terdiri dari tabel-tabel yang mempunyai relasi dari satu tabel ke tabel yang lain. Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

2.15 Part-Of-Speech Tagger (POS Tagger)

Merupakan teknik yang dapat menerapkan penanda tag dari dokumen teks (corpus) dalam beberapa bahasa untuk menjadi kelas kata berdasarkan tipe struktur kalimat dalam dokumen.

Struktur kata ini seperti kata benda (*noun*), kata kerja (*verb*), kata sifat (*adjective*) dan sebagainya. POS Tagger disebut juga grammatical tagging. Proses menandai sebuah kata dalam sebuah teks (corpus) yang sesuai dengan bagian pembicaraan tertentu. Berdasarkan definisi konteks, yaitu hubungan dengan kata-kata yang berdekatan yang terkait dalam frase, kalimat atau paragraf (Daniel Jurafsky, James H. Martin, 2016).

POS Tagging pada zaman dahulu dilakukan menggunakan cara manual yaitu dengan menggunakan jasa ahli bahasa untuk memetakan kelas dari kata dalam corpus. Untuk saat ini POS Tagging dilakukan dalam konteks komputasi linguistik, menggunakan algoritma yang mengasosiasikan istilah diskrit, serta bagian pembicaraan yang tersembunyi (latent). Salah satu pengembangan POS Tagger yang populer saat ini adalah Stanford POS Tagger yang dibangun dalam bahasa Java. Struktur kelas kata POS Tagging digambarkan pada Tabel 2.15-1 (Daniel Jurafsky, James H. Martin, 2016).

Tabel 2.15-1 Struktur Kelas kata POS Tagging

Tag	Deskripsi	Contoh Penerapan
CC	Coordin, conjunction	And, but, or
CD	Cardinal number	One, two
DT	Determiner	A, the
EX	Existential 'there'	There
FW	Foreign word	Mea culpa
IN	Preposition/sub-conjunction	Of, in, by
JJ	Adjective	Yellow
JJR	Adj., comparative	Bigger
JJS	Adj., superlative	Wildest
LS	List item marker	1, 2, one
MD	Modal	Can, should
NN	Noun, sing. Or mass	Llama
NNS	Noun, plural	Llamas
NNP	Proper noun, sing.	IBM
NNPS	Proper noun, plural	Carolinas
PDT	Predeterminer	All, both

POS	Possessive ending	's
PRP	Personal pronoun	I, you, he
PRP\$	Possessive pronoun	Your, one's
RB	Adverb	Quickly, never
RBR	Adverb, comparative	Faster
RBS	Adverb, superlative	Fastest
RP	Participle	Up, off
SYM	Symbol	_, %, &
TO	"to"	to
UH	Interjection	Ah, oops
VB	Verb base form	Eat
VBD	Verb past tense	Ate
VBG	Verb gerund	Eating
VBN	Verb past participle	Eaten
VBP	Verb on-3sg pres	Eat
VBZ	Verb 3-sg pres	Eats
WDT	Wh-determiner	Which, that
WP	Wh-pronoun	What, who
WP\$	Possessive wh-	Whose
WRB	Wh-adverd	How, where
\$	Dollar sign	\$
#	Pound sign	#
"	Left quote	' or "
"	Right quote	' or "
(Left parenthesis	[, (, {, <
)	Right parenthesis],), }, >
,	Comma	,
.	Sentence-final punc	. ! ?
:	Mid-sentence punc	: ; ... -- -

2.16 Bloom Taxonomy

Bloom Taxonomy merupakan sebuah cara untuk membagi cara belajar seseorang menjadi tiga ranah (*domain*) utama. Taksonomi ini dibuat untuk tujuan pendidikan. Dibuat pertama kali oleh Benjamin S. Bloom pada tahun 1956. Tiga ranah (*domain*)

utama diantaranya (Center of Teaching and Learning Stanford University, 1981):

1. Cognitive Domain (ranah kognitif) berisi mengenai perilaku yang menekankan pada aspek intelektual, seperti pengetahuan, pengertian, dan keterampilan berpikir.
2. Affective Domain (ranah afektif) berisi mengenai perilaku yang menekankan pada aspek perasaan dan emosi, seperti minat, sikap, apresiasi, dan cara penyediaan diri.
3. Psychomotor Domain (ranah psychomotor) berisi mengenai aspek ketrampilan motorik seperti tulisan tangan, mengetik, dan mengoperasikan mesin.

Dari ketiga domain diatas, domain yang paling berpengaruh terhadap hasil intelektual adalah domain kognitif. Domain ini selanjutnya dibagi kedalam 6 level, diantaranya (Center of Teaching and Learning Stanford University, 1981):

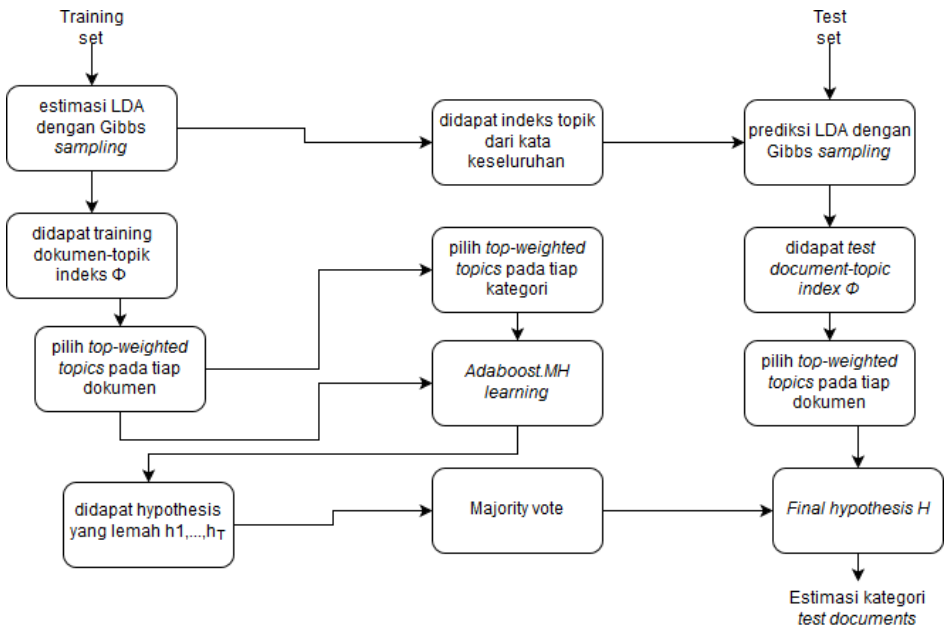
1. Knowledge
Menunjukkan materi yang telah dipelajari sebelumnya dengan mengingat fakta, istilah, konsep dasar dan jawaban.
2. Comprehension
Menunjukkan pemahaman tentang fakta dan gagasan dengan mengatur, membandingkan, menerjemahkan, menafsirkan, memberikan deskripsi dan mengemukakan gagasan utama.
3. Application
Memecahkan masalah dengan menerapkan pengetahuan, fakta, teknik dan peraturan yang diperoleh dengan cara yang berbeda.
4. Analysis
Memeriksa dan memecahkan informasi menjadi beberapa bagian dengan mengidentifikasi motif atau sebab, serta membuat kesimpulan dan menemukan bukti untuk mendukung generalisasi.
5. Synthesis
mengkompilasi informasi bersama-sama dengan cara yang berbeda dengan menggabungkan elemen dalam pola baru atau mengusulkan solusi alternatif.

6. Evaluation

Menyajikan dan mempertahankan pendapat dengan membuat penilaian tentang informasi, validitas gagasan atau kualitas kerja berdasarkan seperangkat kriteria.

2.17 Metode *LDA-AdaBoost.MH*

LDA-AdaBoost.MH adalah sebuah *multilabel text classifier* yang didasari oleh *AdaBoost.MH algorithm*. Dalam metode ini representasi teks per kata yang seharusnya menggunakan *Bag-Of-Words* diganti dengan representasi teks *latent topics* menggunakan *Latent Dirichlet Allocation*.



Gambar 2.17.1 Proses Klasifikasi *LDA-AdaBoost.MH*

Pada Gambar 2.17 diatas, dijelaskan bagaimana alur *learning* dan klasifikasi dari *LDA-AdaBoost.MH*. Pertama,

training set di preproses (*tokenization*, *normalization* dan *stop words removal*) kemudian LDA menyimpulkan topik latennya dibantu estimasi dari *Gibbs sampling*.

Dalam *LDA-AdaBoost.MH* ini, jumlah topik yang digunakan adalah sesuai dengan Rumus (2) berikut:

$$\text{jumlah topik} = \frac{\text{jumlah dokumen}}{\text{jumlah kategori}} \quad (2)$$

Tujuan utama LDA dalam metode ini adalah untuk mengelompokkan kata-kata yang sama secara semantik ke dalam beberapa topik tersebut. Rumus (2) diatas ditujukan untuk memberikan relasi antara jumlah dokumen dan kategori, jumlah topik, serta jumlah topik sesuai proporsi ukuran dataset dan jumlah kategori.

Dalam tiap dokumen pada dokumen-topik indeks distribusi, ekstraksi topik dilakukan jika *weights* lebih besar dari *threshold*. Threshold dikalkulasi dengan merata-rata semua *weights* dari topik tiap-tiap dokumen, seperti pada Rumus (3) berikut:

$$\text{Threshold}_{\text{topic per document}} = \frac{\sum \text{weight}_{\text{topic per document}}}{\text{jumlah topik}} \quad (3)$$

Kemudian untuk tiap kategori, pilih topik yang *weightnya* melebihi dari *threshold*. *Threshold* tersebut adalah rata-rata dari penjumlahan semua *weights* dari topik tiap-tiap dokumen dalam kategori, seperti pada Rumus (4) berikut:

$$\text{Threshold}_{\text{topic per document}} = \frac{\sum_m \sum_n \text{weight}_{\text{topic per document}}}{\text{jumlah topik}} \quad (4)$$

m = jumlah dokumen per kategori dan n = jumlah topik per dokumen.

Terakhir, inti dari langkah-langkah diatas adalah untuk *learning* dan evaluasi dari *AdaBoost.MH*, tiap dokumen akan diwakili oleh jumlah topik yang kemudian diekstrak dan dipilih sesuai dengan *weights* yang diinginkan (melebihi *threshold*).

2.18 Macam-macam Pengembangan Metode LDA

Metode Latent Dirichlet Allocation (LDA) merupakan salah satu metode topik modeling yang mudah untuk dikembangkan. Penambahan terhadap variabel penilaian, aspek keakuratan, serta kondisi kasus yang berkembang merupakan beberapa hal yang mengharuskan terus adanya perkembangan dalam sebuah metode *machine learning*. Keunggulan dan hasil yang lebih bagus menjadi target yang harus dikejar oleh akademisi pengembang. Beberapa versi metode LDA yang sudah ada diantaranya sebagai berikut:

1. Regular Latent Dirichlet Allocation (LDA) (David M. Blei, Andrew Y. Ng, Michael I. Jordan, 2003)
2. Supervised Latent Dirichlet Allocation (sLDA) (David M. Blei, Jon D. McAuliffe, 2007)
3. Discriminative Latent Dirichlet Allocation (DiscLDA) (Simon Lacoste-Julien, Fei Sha, Michael I. Jordan, 2008)
4. Mixture Latent Dirichlet Allocation (MM-LDA) (Nikhil Johri, Daniel Ramage, Daniel A. McFarland, Daniel Jurafsky, 2011)
5. Latent Dirichlet Allocation with AdaBoost.MH (LDA-AdaBoost.MH) (Daniel Ramage, David Hall, Ramesh Nallapati, Christopher D. Manning, 2009)

2.19 Term Frequency – Invers Document Frequency (TF-IDF)

Term Frequency – Inver Document Frequency atau biasa disebut dengan TF-IDF merupakan sebuah metode untuk mendapatkan bobot sebuah term dari sebuah corpus. Metode ini terkenal efisien, mudah dan memiliki hasil yang akurat. TF-IDF didapatkan dengan mencari nilai Term Frequency (TF) dan nilai Invers Documen Frequency (IDF) yang terlebih dahulu mencari nilai Document Frequency (DF) (Juan Ramos, 2003).

2.19.1 Term Frequency (TF)

Merupakan jumlah kemunculan suatu term dalam dokumen. Nilai TF kemudian dilogaritmikkan untuk mengurangi besarnya bilangan. Persamaan mencari TF seperti persamaan (5).

$$Tf = tfdt \quad (5)$$

2.19.2 Invers Document Frequency (IDF)

Sebelum mencari IDF terlebih dahulu mencari nilai DF yaitu jumlah dokumen dimana sebuah term muncul. Setelah mendapatkan nilai DF kemudian menghitung nilai IDF yang berfungsi mengurangi bobot suatu term jika kemunculannya banyak tersebar di seluruh koleksi dokumen (Juan Ramos, 2003). Persamaan DF seperti pada persamaan (6), dan IDF pada persamaan (7).

$$Df = dft \quad (6)$$

$$Idf_t = \log_{10} \left(\frac{N}{dft} \right) \quad (7)$$

Keterangan:

- Tf = term frequency
- Df = dokumen frequency
- IDF = invers dokumen requency
- N = jumlah dokumen

2.20 Fase Teks Mining (Metode LDA-AdaBoost.MH) Studi Kasus Pemingkatan Perguruan Tinggi

Terdapat 6 tahap utama dalam alur metode *LDA-AdaBoost.MH* dari dokumen mentah hingga menjadi bobot final reputasi akademik perguruan tinggi, diantaranya sebagai berikut:

1. Tahap preprocessing

Tahap preprocessing merupakan proses untuk membersihkan kata (*term*), dimana kumpulan dokumen abstrak paper yang terdiri dari banyak dokumen akan dikelola diambil kata bakunya saja. Kata baku menunjukkan komposisi keberadaan kata (*term*) dalam dokumen (Steven Bird, 2006). Tahap preprocessing terdiri dari beberapa sub-proses diantaranya:

1.1. Case folding

Merupakan proses merubah huruf bertipe besar (*uppercase*) menjadi bertipe kecil (*lowercase*) dalam sebuah corpus dokumen.

1.1. Tokenisasi

Merupakan proses pemecahan kumpulan kalimat menjadi kata (*term*) yang berdiri sendiri-sendiri (satu suku kata).

1.2. Remove punctuation

Merupakan proses penghapusan karakter yang dapat disesuaikan pengguna. Pada umumnya karakter yang dihapus adalah karakter non-huruf (tanda baca - angka).

1.3. Stopwords

Merupakan proses penghapusan kata (*term*) yang tidak begitu penting keberadaanya didalam dokumen. Sehingga dapat memaksimalkan keberadaan kata yang penting.

1.4. Stemming

Merupakan proses pemotongan imbuhan (*prefix*, *suffix*, *infix*) dari sebuah kata (*term*) dalam sebuah dokumen.

Hasil dari tahap preprocessing berupa kumpulan kata baku dan telah bersih kemudian hasil kata ini akan menjadi bahan pairing dalam pencocokan Term Frequency (TF) untuk proses Menentukan asumsi label awal dokumen.

2. Tahap memperkaya corpus TKT

Corpus TKT merupakan dokumen kumpulan kata kunci (*keyword*) yang merepresentasikan kelas dari *term* pada dokumen dataset untuk masuk kedalam topik tingkat kesiapterapan teknologi (TKT) yang mempunyai 9 kelas. Corpus TKT mengadopsi corpus Taxonomy Bloom yang terdiri dari 6 kelas (Center of Teaching and Learning Stanford University, 1981). Untuk dapat digunakan sebagai corpus dalam teks mining metode LDA-AdaBoost.MH, kelas corpus dalam Taxonomy Bloom perlu dipecah menjadi 9 kelas sesuai kelas TKT (Kementerian, Riset Teknologi dan Pendidikan Tinggi, 2016). Breakdown menggunakan pengurutan aspek

konten dari setiap *keyword*, dimana *keyword* yang mempunyai apek tipe dasar berada pada level bawah dan aspek tipe penerapan berada pada level atas.

Selain menggunakan breakdown terhadap 6 corpus taxonomy Bloom menjadi 9 corpus TKT, dalam Tugas Akhir ini dilakukan juga pengembangan corpus dari beberapa cara. Cara pertama mengembangkan berdasarkan adopsi teknik POS Tagging dokumen abstrak. POS tagging akan memecah sebuah kalimat untuk diambil kelas tipe kata seperti verb, noun, adjective, adverb dan sebagainya (Daniel Jurafsky, James H. Martin, 2016). Sebelm melakukan POS Tagging, dokumen abstrak paper akan dilakukan pencocokan terhadap *keyword* di masing-masing corpus TKT. setelah melakukan pencocokan dan ditemukan kecocokan, selanjutnya akan dicek terlebih dahulu kata (*term*) tersebut tergolong pada kelas kata apa dari prinsip POS Tagging, setelah diketahui kelas tipe kata kemudian akan mengakuisisi kata (*term*) lain yang berdekatan dengan kata (*term*) tersebut dengan jangkauan akuisisi beragam, mulai dari akuisisi 1 hingga 10. Dalam percobaan selanjutnya acuan menggunakan akuisisi 1. Kelas kata (*term*) hasil akuisisi akan masuk pada corpus yang sama dengan kata yang dicocokkan sebelumnya.

Untuk cara kedua mengembangkan berdasarkan melihat sinonim dari setiap kata yang ada pada 9 corpus TKT. Setiap kata (*term*) akan dicari sinonimnya menggunakan library WordNet dalam bahasa pemrograman python (Tingting Wei, Yonghe Lu, Houyou Chang, Qiang Zhou, Xianyu Bao, 2015). Terdapat 4 tipe percobaan dalam menghasilkan corpus baru dengan mengacu pada pengembangan corpus dari segi sinonim, percobaan pertama dengan melihat sinonim level 1 dari setiap kata yang ada pada 9 corpus TKT. sinonim level 1 didapatkan dari satu nilai similaritas paling tinggi antara sinonim. Percobaan kedua dengan melihat sinonim level 2 dari setiap kata. Sinonim level 2 didapatkan dari dua nilai similaritas tertinggi. Percobaan 3 menggunakan prinsip yang sama hanya saja merubahnya

menjadi tiga nilai similaritas tertinggi. Sedangkan yang terakhir adalah menggunakan sinonim level all, dimana dalam percobaan ini akan diambil semua sinonim dari setiap kata yang ada pada 9 corpus TKT. hasil dari pencarian sinonim akan ditambahkan pada corpus TKT final.

Dalam Tugas Akhir ini disimpulkan menggunakan proses memperkaya corpus TKT dengan cara mengembangkan dari segi sinonim pada percobaan sinonim level 3 dan all sebagai perbandingan. Karena berdasar hasil analisa akhir menunjukkan bahwa distance atau bisa disebut hasil akhir dari kedua percobaan ini lebih bagus dari percobaan lainnya.

3. Tahap penentuan asumsi topik awal menggunakan Term Frequency (TF)

Tahap ini dilakukan dengan menghitung kemunculan term paling banyak dari dokumen abstrak *paper* pada 9 corpus TKT final (Juan Ramos, 2003). 9 corpus TKT final yang digunakan berasal dari beberapa percobaan dalam tahap sebelumnya, diantaranya menggunakan corpus TKT hasil memperkaya menggunakan teknik POS Tagging akuisisi 1, percobaan sinonim WordNet 1, percobaan sinonim WordNet 2, percobaan sinonim WordNet 3, dan percobaan sinonim WordNet all. Dari semua percobaan ini disimpulkan bahwa menggunakan percobaan 3 dan all lebih bagus dalam menghasilkan distance hasil akhir.

Dokumen yang mempunyai kecocokan berdasarkan term frequency (TF) akan dicatat jumlah term yang sesuai, dan diurutkan berdasarkan jumlah kata yang paling tinggi hingga yang paling rendah. Kecocokan ini menandakan bahwa dokumen relevan berdasarkan keyword dengan ke-9 corpus TKT. 3 kecocokan teratas antara dokumen dan corpus TKT akan digunakan sebagai asumsi label awal dokumen untuk masukan tahap inferensi Gibbs Sampling LDA-AdaBoost.MH.

4. Tahap generative model LDA-AdaBoost.MH dan inferensi Gibbs Sampling

Tahap ini merupakan proses ini untuk menghasilkan probabilitas akhir setiap label yang menjadi topik final. Masukan pada tahap ini berupa asumsi label topik yang diharuskan lebih dari satu label beserta dokumen abstrak paper. Asumsi label topik ini akan diproses menggunakan threshold, iterasi dan ketentuan variabel masukan lainnya yang ditentukan oleh user (Daniel Ramage, David Hall, Ramesh Nallapati, Christopher D. Manning, 2009). Penggunaan metode LDA-AdaBoost.MH berguna untuk memastikan asumsi label mempunyai probabilitas tertinggi agar berada pada satu label saja dari sebuah dokumen uji. Hasil keluaran proses LDA-AdaBoost.MH berupa probabilitas term dari dokumen abstrak dengan asumsi label topik, serta probabilitas dokumen abstrak dengan asumsi label topik. Dalam Tugas Akhir ini, keluaran yang berperan penting untuk proses selanjutnya adalah probabilitas dokumen abstrak dengan label topik.

5. Tahap penentuan label topik final dan pembobotan

Pada tahap ini telah didapatkan topik final dari asumsi label topik yang dijadikan batasan uji Gibbs Sampling. Masing-masing asumsi label topik ini berguna untuk proses pembobotan. Setiap asumsi label topik hasil keluaran LDA-AdaBoost.MH beserta probabilitas akan dihitung sebagai masukan tahap penghitungan skor final. Hasil penghitungan ini akan mencerminkan skor kriteria reputasi akademik dari perguruan tinggi.

2.21 Akreditasi Perguruan Tinggi

Akreditasi perguruan tinggi merupakan salah satu cara untuk menilai kualitas perguruan tinggi secara keseluruhan, yang bertujuan untuk mengetahui komitmen perguruan tinggi terhadap kapasitas dan efektifitas pendidikan yang diselenggarakan, yang didasarkan pada standar akreditasi yang telah ditetapkan oleh

badan akreditasi nasional (Badan Akreditasi Nasional Perguruan Tinggi, 2011).

Di Indonesia pihak yang berwenang untuk melakukan akreditasi institusi perguruan tinggi adalah Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT). Akreditasi ini dilakukan oleh tim asesor yang turun langsung kepada perguruan tinggi terkait untuk dinilai dari beberapa hal. Berikut adalah poin-poin penting dalam penilaian akreditasi institusi oleh BAN-PT (Badan Akreditasi Nasional Perguruan Tinggi, 2011).

1. Menilai kesesuaian visi, misi, tujuan dan sasaran, serta strategi pencapaian dalam perguruan tinggi dengan kondisi yang sudah ada sekarang.
2. Menilai tata pamong (kepengurusan), kepemimpinan, sistem pengelolaan, dan penjaminan mutu.
3. Menilai kualitas mahasiswa dan lulusan.
4. Menilai sumber daya manusia (tenaga pengajar dan administrasi).
5. Menilai kesesuaian kurikulum, model pembelajaran, serta suasana akademik.
6. Menilai kebutuhan pembiayaan, sarana dan prasarana, serta sistem informasi dalam perguruan tinggi.
7. Menilai tentang penelitian, pelayanan/pengabdian masyarakat, serta kerjasama.

Penilaian akreditasi ini dilakukan dengan cara survey langsung terhadap perguruan tinggi, sehingga data yang didapatkan dapat objektif terhadap perguruan tinggi tersebut. Akreditasi yang dilakukan oleh BAN-PT tersebut mempunyai 2 jenis, yaitu akreditasi institusi dan akreditasi profesi.

Akreditasi institusi yang merupakan penilaian kualitas institusi dengan memperhatikan poin-poin penilaian diatas. Sedangkan akreditasi profesi merupakan penilaian kualitas terhadap bidang studi/departemen/jurusan dalam perguruan tinggi tersebut, yang mana akan menentukan kualitas untuk akreditasi institusi juga. Berikut adalah score penilaian akreditasi BAN-PT.

- a) Nilai 601 – 700, mendapatkan peringkat A
- b) Nilai 501 – 600, mendapatkan peringkat B

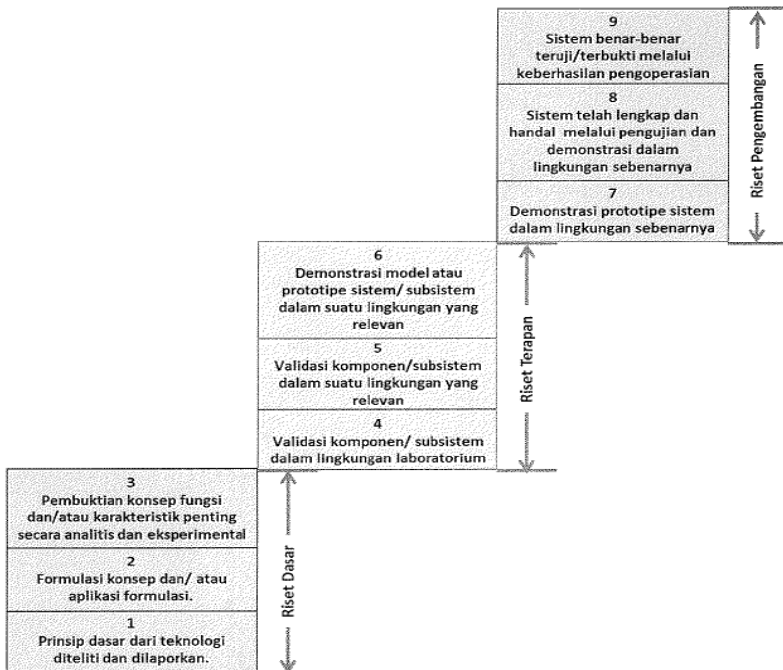
- c) Nilai 401 – 500, mendapatkan peringkat C
- d) Nilai 0 – 400, mendapatkan peringkat N/A (not applicable)

Dalam Tugas Akhir ini, salah satu kriteria penilaian yang menjadi acuan adalah kriteria reputasi employer. Dengan menggunakan kriteria akreditasi perguruan tinggi oleh BAN-PT yang dijelaskan pada poin penilaian kualitas mahasiswa dan lulusan sudah mengandung kriteria reputasi employer yang dibutuhkan.

2.22 *Tingkat Kesiapterapan Teknologi (TKT)*

Tingkat Kesiapterapan Teknologi atau yang biasa disebut dengan TKT merupakan tingkat kondisi kematangan atau kesiapterapan suatu hasil penelitian (*research*) dan pengembangan prototype tertentu yang selanjutnya akan masuk pada tahap komersialisasi, yang diukur secara sistematis dengan tujuan untuk dapat dimanfaatkan oleh pengguna, baik oleh pemerintah, maupun masyarakat luas (Kementerian, Riset Teknologi dan Pendidikan Tinggi, 2016).

Setiap jenis penelitian (*research*) dan pengembangan prototype dikelompokkan menjadi 9 tingkatan dengan masing-masing memiliki indikator yang berbeda-beda. Seperti pada Gambar 2.22-1.



Gambar 2.22-1 Hirarki Tingkat Kesiapterapan Teknologi (TKT)

Berdasarkan hirarki tingkatan diatas, dapat dikelompokkan menjadi 3 kelompok tingkatan besar (Kementerian, Riset Teknologi dan Pendidikan Tinggi, 2016), diantaranya.

1. Riset dasar untuk TKT tingkat 1-3

Dalam kelompok ini, penelitian yang dimaksud adalah penelitian yang berfokus pada pembahasan metode, pengembangan teori, pembahasan gagasan produk. Penelitian ini akan bertujuan untuk membuat produk yang bermanfaat bagi masyarakat.

2. Riset terapan untuk TKT tingkat 4-6

Dalam kelompok ini, penelitian yang dimaksud adalah penelitian yang berfokus pada yang menghasilkan produk, demonstrasi model atau prototype subsistem dalam lingkungan yang relevan, dan berfokus pada validasi dalam lingkungan laboratorium. Penelitian ini sudah membuat produk/prototype.

3. Riset pengembangan untuk TKT tingkat 7-9

Dalam kelompok ini, penelitian yang dimaksud adalah penelitian yang berfokus pada pengembangan produk yang sudah ada sehingga membuat riset pendukung baru untuk produk yang sudah ada.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis permasalahan dan perancangan Tugas Akhir. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan Tugas Akhir sekaligus solusi yang ditawarkan oleh penulis. Analisis kebutuhan berisi kebutuhan-kebutuhan yang diperlukan perangkat lunak. Kemudian dilanjutkan dengan perancangan sistem yang dibuat. Perancangan direpresentasikan dengan diagram UML (*Unified Modelling Language*).

3.1. Analisis

Tahap analisis dibagi menjadi beberapa bagian antara lain cakupan permasalahan, deskripsi umum sistem, kasus penggunaan sistem, dan kebutuhan perangkat lunak.

3.1.1 Cakupan permasalahan

Permasalahan utama yang diangkat dalam pembuatan Tugas Akhir ini adalah bagaimana mengimplementasikan suatu aplikasi berbasis web yang bertujuan untuk memberikan peringkat universitas se-Indonesia mengacu pada *QS World University Rankings*.

Aplikasi berbasis web ini mengganti beberapa indikator *QS World University Rankings* yang didapatkan secara manual dengan indikator yang berbeda namun tidak mengurangi akurasi peringkat serta dapat diperoleh secara otomatis.

Untuk menangani hal tersebut, perlu dibangun aplikasi berbasis web menggunakan metode *LDA-AdaBoost.MH* untuk memberikan informasi peringkat universitas yang lebih efisien dan otomatis.

3.1.2 Deskripsi umum sistem

Pada Tugas Akhir ini dibangun aplikasi pemeringkatan universitas di Indonesia menggunakan metode *LDA-AdaBoost.MH* yang terintegrasi ke dalam sebuah web dengan alamat *rankinguniversity.info*. Tujuan dari aplikasi pemeringkatan universitas di Indonesia menggunakan metode *LDA-AdaBoost.MH*

adalah untuk memberikan informasi peringkat universitas menggunakan indikator yang didapatkan secara otomatis dan dapat diakses sewaktu-waktu secara *up-to-date*.

3.1.3 Aktor

Pada sistem yang akan dibangun, aktor yang akan menjadi pengguna sistem adalah administrator dibalik sistem yang berjalan dan seluruh kalangan masyarakat yang memiliki keperluan dalam pemeringkatan universitas di Indonesia. Administrator mempunyai hak ases dalam pengelolaan data indikator-indikator yang digunakan dalam proses pemeringkatan yang terdiri atas membuat (*create*), memperbarui (*update*) dan menghapus (*delete*). Sedangkan masyarakat mempunyai hak akses melihat (*view*) data yang telah dikelola oleh administrator.

3.1.4 Spesifikasi Kebutuhan Sistem

Pada perangkat lunak pemeringkatan perguruan tinggi di Indonesia untuk mengukur tingkat kesiapterapan teknologi (TKT) menggunakan metode *LDA-AdaBoost.MH* ini mempunyai spesifikasi kebutuhan yang mencakup kebutuhan fungsional sistem. Kebutuhan fungsional berisikan proses-proses yang dibutuhkan dalam sistem dan harus dijalankan agar menghasilkan pemeringkatan yang akurat terhadap *Ground Truth* yang telah ditetapkan. Kebutuhan fungsional tersebut dideskripsikan dalam Tabel 1-1.

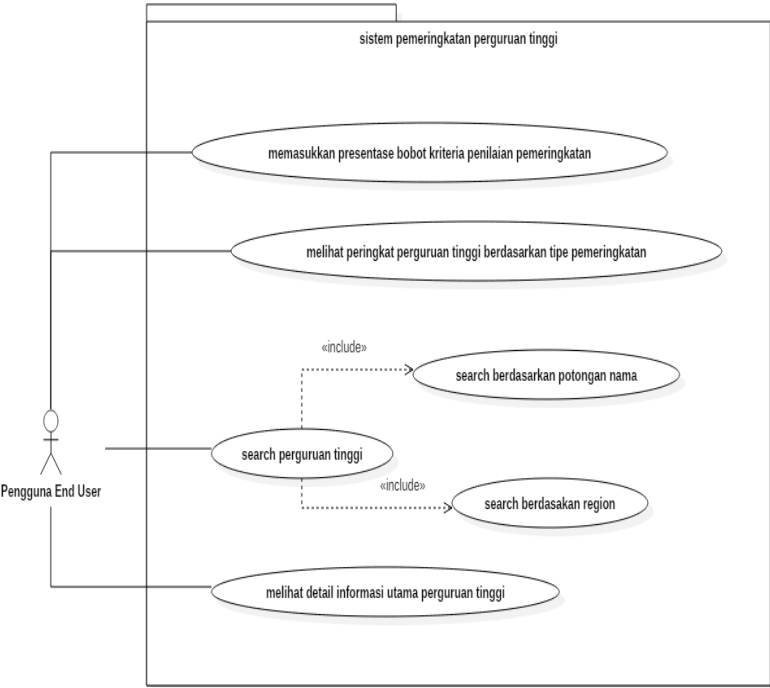
Tabel 1-1 Daftar Kebutuhan Fungsional Sistem

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi	Aktor
F-001	Memasukkan presentase bobot kriteria penilaian pemeringkatan	Pengguna dapat memasukkan presentase bobot pemeringkatan sesuai dengan kebutuhan.	End User
F-002	Melihat peringkat perguruan tinggi	Pengguna dapat melihat peringkat perguruan tinggi dengan menggunakan bobot yang	End User

	berdasarkan tipe pemeringkatan	dipilih sesuai dengan tipe pemeringkatan.	
F-003	Mencari perguruan tinggi dari database berdasarkan potongan nama	Pengguna dapat mencari perguruan tinggi dengan memasukkan keyword berupa potongan nama perguruan tinggi dalam kotak pencarian search box.	End User
F-004	Mencari perguruan tinggi dari database berdasarkan region wilayah provinsi	Pengguna dapat mencari perguruan tinggi dengan memilih region wilayah provinsi yang disediakan dalam map peta Indonesia dalam 33 provinsi.	End User
F-005	Melihat detail informasi akademik perguruan tinggi	Pengguna dapat melihat informasi detail dari perguruan tinggi, yang dapat dikelompokkan menjadi 6 kategori, diantaranya kepopuleran webhost, mahasiswa, akreditasi, produktifitas institusi, kecukupan dosen pengajar, dan kualitas akademisi.	End User

3.1.5 Kasus Penggunaan

Kasus penggunaan yang dibutuhkan pada sistem sesuai dengan analisa yang sudah dilakukan. Diagram kasus penggunaan yang mewakili kebutuhan sistem dapat dilihat pada Gambar 1-1 dan kode kasus penggunaan pada Tabel 1-2.



Gambar 1-1 Diagram Kasus Penggunaan

Tabel 1-2 Daftar Kebutuhan Fungsional Sistem

Kode Kasus Penggunaan	Kasus Penggunaan
UC-001	Memasukkan presentase bobot kriteria penilaian pemeringkatan
UC-002	Melihat peringkat perguruan tinggi berdasarkan tipe pemeringkatan
UC-003	Mencari perguruan tinggi berdasarkan potongan nama
UC-004	Mencari perguruan tinggi berdasarkan region wilayah provinsi

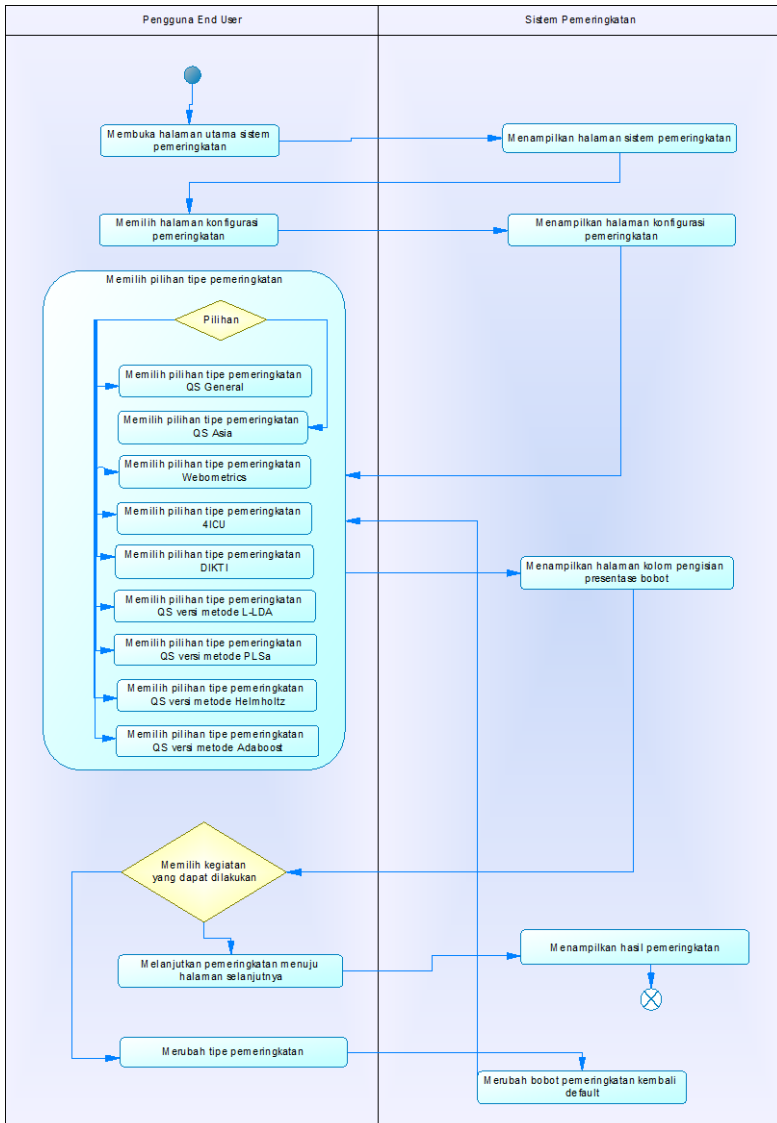
UC-005	Melihat detail informasi utama perguruan tinggi
--------	---

3.1.5.1 UC-001 Memasukkan Presentase Bobot Kriteria Penilaian Pemeringkatan

Tabel 1-3 Case Memeasukkan Presentase Bobot Kriteria Penilaian Pemeringkatan

Nama	Memasukkan Presentase Bobot Kriteria Penilaian Pemeringkatan
Nomor	UC-001
Deskripsi	Kasus penggunaan ini digunakan untuk menentukan bobot pemeringkatan pada tipe pemeringkatan versi metode LLDA dengan menentukan bobot pemeringkatan disetiap kriteria yang diinginkan oleh pengguna
Tipe	Fungsional
Aktor	Pengguna End User
Kondisi Awal	Pengguna sudah masuk dalam sistem
Kondisi Akhir	Sistem menampilkan halaman pemilihan kriteria dan tipe pemeringkatan yang sudah mempunyai bobot
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna End User membuka halaman utama sistem pemeringkatan 2. Sistem menampilkan halaman utama sistem pemeringkatan 3. Pengguna memilih halaman konfigurasi pemeringkatan dengan melakukan scroll kursor kebawah 4. Sistem menampilkan halaman konfigurasi pemeringkatan 5. Pengguna End User memilih salah satu pilihan tipe pemeringkatan yang terdapat pada halaman konfigurasi pemeringkatan <ol style="list-style-type: none"> a. Tipe pemeringkatan QS General b. Tipe pemeringkatan QS versi Asia c. Tipe pemeringkatan Webometrics d. Tipe pemeringkatan 4ICU e. Tipe pemeringkatan DIKTI f. Tipe pemeringkatan QS versi metode LLDA g. Tipe pemeringkatan QS versi metode PLSA h. Tipe pemeringkatan QS versi metode Helmholtz i. Tipe pemeringkatan QS versi metode Adaboost LDA

	6. Sistem menampilkan halaman kolom pengisian presentase bobot pada tipe pemeringkatan yang dipilih 7. Pengguna End User memilih kegiatan yang dapat dilakukan j. Melanjutkan pemeringkatan menuju halaman selanjutnya k. Merubah tipe pemeringkatan
Alur Alternatif	-

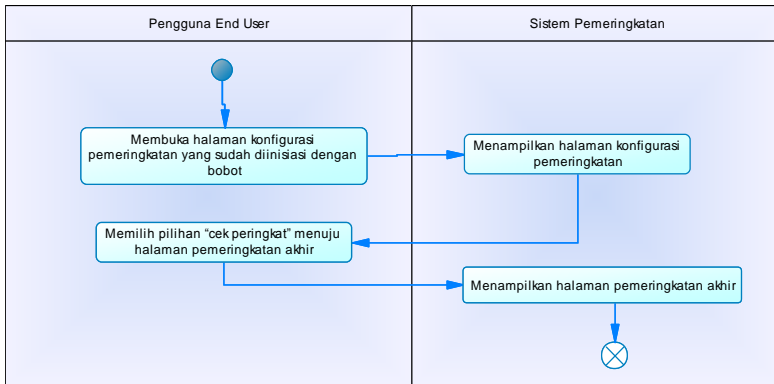


Gambar 1-2 Diagram Aktivitas Use Case UC-001

3.1.5.2 UC-002 Melihat Peringkat Perguruan Tinggi Berdasarkan Tipe Pemeringkatan

Tabel 1-4 Use Case Melihat Peringkat Perguruan Tinggi Berdasarkan Tipe Pemeringkatan

Nama	Melihat Peringkat Perguruan Tinggi Berdasarkan Tipe Pemeringkatan
Nomor	UC-002
Deskripsi	Kasus penggunaan ini digunakan untuk melihat pemeringkatan akhir perguruan tinggi
Tipe	Fungsional
Aktor	Pengguna End User
Kondisi Awal	Pengguna sudah masuk dalam sistem dan memasukkan bobot kriteria penilaian
Kondisi Akhir	Sistem menampilkan halaman akhir yang menampilkan peringkat perguruan tinggi sesuai dengan bobot dan pilihan tipe pemeringkatan yang diinisiasi sebelumnya
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna End User membuka halaman konfigurasi pemeringkatan yang sudah diinisiasi dengan bobot 2. Sistem menampilkan halaman konfigurasi pemeringkatan yang sudah diinisiasi dengan bobot 3. Pengguna End User memilih pilihan “cek peringkat” menuju halaman pemeringkatan akhir 4. Sistem menampilkan halaman pemeringkatan akhir
Alur Alternatif	-



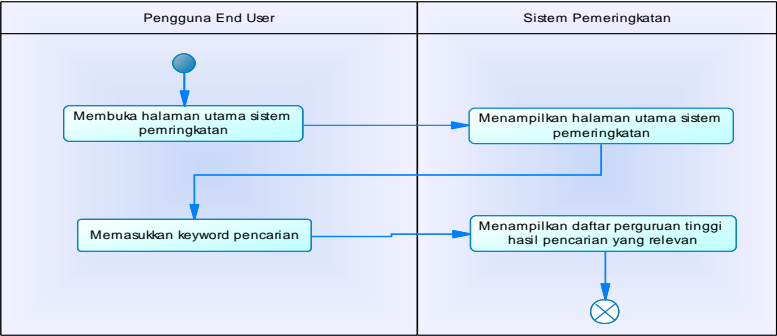
Gambar 1-3 Diagram Aktivitas Use Case UC-002

3.1.5.3 UC-003 Mencari Perguruan Tinggi Berdasarkan Potongan Nama

Tabel 1-5 Use Case Mencari Perguruan Tinggi Berdasarkan Potongan Nama

Nama	Mencari Perguruan Tinggi Berdasarkan Potongan Nama
Nomor	UC-003
Deskripsi	Kasus penggunaan ini digunakan untuk mencari informasi perguruan tinggi berdasarkan masukan berupa potongan nama perguruan tinggi pada kolom search pada halaman utama
Tipe	Fungsional
Aktor	Pengguna End User
Kondisi Awal	Pengguna sudah masuk dalam sistem
Kondisi Akhir	Sistem menampilkan halaman hasil pencarian berdasarkan keyword potongan nama yang dimasukkan dalam kolom search yang berisikan list perguruan tinggi yang relevan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna End User membuka halaman utama sistem pemeringkatan 2. Sistem menampilkan halaman utama sistem pemeringkatan 3. Pengguna End User memasukkan keyword pencarian berupa potongan nama pada kolom pencarian

	4. Sistem menampilkan daftar perguruan tinggi hasil pencarian yang relevan dalam halaman result
Alur Alternatif	-



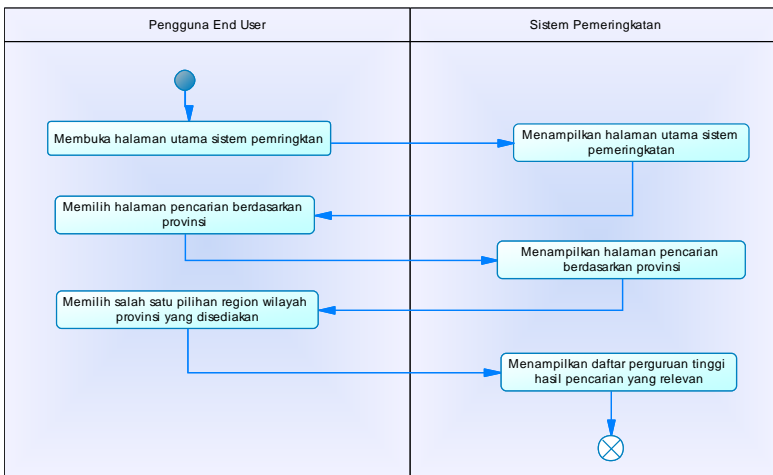
Gambar 1-4 Diagram Aktifitas Use Case UC-003

3.1.5.4 UC-004 Mencari Perguruan Tinggi Berdasarkan Region Wilayah Provinsi

Tabel 1-6 Use Case Mencari Perguruan Tinggi Berdasarkan Region Wilayah Provinsi

Nama	Mencari Perguruan Tinggi Berdasarkan Region Wilayah Provinsi
Nomor	UC-004
Deskripsi	Kasus penggunaan ini digunakan untuk mencari informasi perguruan tinggi berdasarkan kategori region wilayah provinsi perguruan tinggi pada halaman pencarian provinsi
Tipe	Fungsional
Aktor	Pengguna End User
Kondisi Awal	Pengguna sudah masuk dalam sistem
Kondisi Akhir	Sistem menampilkan halaman hasil pencarian berdasarkan region wilayah provinsi yang dipilih pada map provinsi
Alur Normal	1. Pengguna End User membuka halaman utama sistem peningkatan

	<ol style="list-style-type: none"> 2. Sistem menampilkan halaman utama sistem pemeringkatan 3. Pengguna End User memilih halaman pencarian berdasarkan provinsi dengan melakukan scroll kursor kebawah 4. Sistem menampilkan halaman pencarian berdasarkan provinsi 5. Pengguna End User memilih salah satu pilihan region wilayah provinsi yang disediakan 6. Sistem menampilkan daftar perguruan tinggi yang berada pada region wilayah provinsi yang dipilih
Alur Alternatif	-



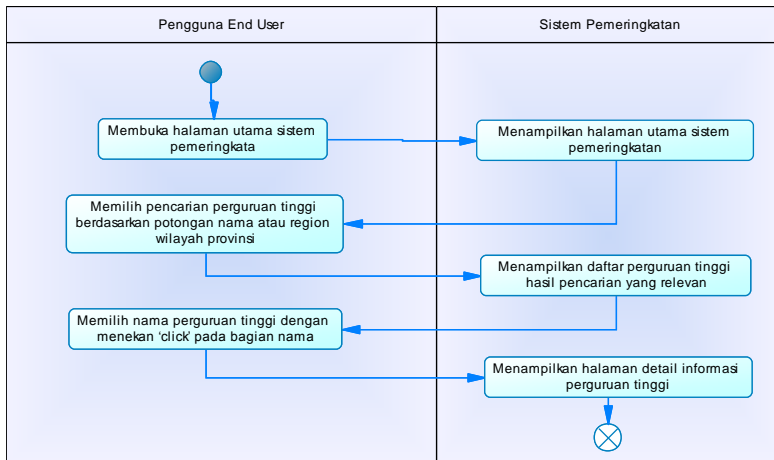
Gambar 1-5 Diagram Aktifitas Use Case UC-004

3.1.5.5 UC-005 Melihat Detail Informasi Utama Perguruan Tinggi

Tabel 1-7 Use Case Melihat Detail Informasi Utama Perguruan Tinggi

Nama	Melihat Detail Informasi Utama Perguruan Tinggi
Nomor	UC-005

Deskripsi	Kasus penggunaan ini digunakan untuk melihat detail informasi yang dimiliki perguruan tinggi, meliputi kepopuleran webhost, mahasiswa, status akreditasi, produktifitas institusi, kecukupan dosen pengajar, dan kualitas akademisi serta dilengkapi dengan grafik tipe pemeringkatan yang dimiliki
Tipe	Fungsional
Aktor	Pengguna End User
Kondisi Awal	Pengguna sudah melakukan pencarian perguruan tinggi berdasarkan potongan nama atau region wilayah provinsi dan sistem menampilkan daftar perguruan tinggi
Kondisi Akhir	Sistem menampilkan halaman detail informasi yang mengandung banyak informasi, diantaranya kepopuleran webhost, mahasiswa, status akreditasi, produktifitas institusi, kecukupan dosen pengajar, dan kualitas akademisi serta dilengkapi dengan grafik tipe pemeringkatan yang dimiliki
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna End User membuka halaman utama sistem pemeringkatan 1. Sistem menampilkan halaman utama sistem pemeringkatan 2. Pengguna End User memilih pencarian perguruan tinggi berdasarkan potongan nama atau region wilayah provinsi 3. Sistem menampilkan daftar perguruan tinggi hasil pencarian yang relevan pada halaman pencarian 4. Pengguna End User memilih nama perguruan tinggi dengan menekan 'click' pada bagian nama 5. Sistem menampilkan halaman detail informasi perguruan tinggi
Alur Alternatif	-



Gambar 1-6 Diagram Aktivitas Use Case UC-005

3.1.8 Label Kelas dalam Teks Mining Metode LDA-AdaBoost.MH

Penggunaan metode LDA-AdaBoost.MH yang bersifat supervised mengharuskan adanya keberadaan kelas atau bisa disebut juga dengan label. Label ini digunakan sebagai penanda pengetahuan dokumen masuk kedalam kondisi seperti apa. Kondisi dalam studi kasus ini adalah hirarki dari Tingkat Kesiapterapan Teknologi yang berjumlah 9 level. Berikut adalah daftar kelas yang digunakan dalam Tugas Akhir ini.

Tabel 1-8 Penjelasan Setiap Kelas Label

No.	Hirarki TKT	Kelas/Label	Keterangan
1	Level 1	T1	Prinsip dasar dari teknologi diteliti dan dilaporkan.
2	Level 2	T2	Formulasi konsep dan/atau aplikasi formulasi.
3	Level 3	T3	Pembuktian konsep fungsi dan/atau karakteristik penting secara analisa dan eksperimental
4	Level 4	T4	Validasi komponen atau subsistem dalam lingkup laboratorium
5	Level 5	T5	Validasi komponen atau subsistem dalam lingkungan yang relevan
6	Level 6	T6	Demonstrasi model atau prototype sistem atau subsistem dalam lingkungan yang relevan
7	Level 7	T7	Demonstrasi model dalam lingkungan yang sebenarnya
8	Level 8	T8	Sistem telah lengkap dan handal melalui pengujian dan demonstrasi dalam lingkungan sebenarnya

9	Level 9	T9	Sistem benar-benar teruji melalui keberhasilan pengoperasian
---	---------	----	--

3.1.9 Ground Truth Pengujian Sistem

3.1.9.1 Peringkat QS World University Rankings

Dalam Tugas Akhir ini pemeringkatan keseluruhan dengan menggunakan 6 indikator penilaian mengacu pada kebenaran peringkat QS World University Rankings tahun 2016-2017 sesuai dengan waktu pengerjaan Tugas Akhir ini dimulai. Selain itu melakukan uji coba perbandingan jika mengacu pada Ground Truth peringkat QS World University Rankings dari beberapa tahun yang berbeda. Berikut adalah beberapa peringkat QS World University Rankings dan peringkat reputasi akademik dari berbagai jangka waktu.

1. QS 2015-2015

Peringkat universitas wilayah Indonesia berdasarkan QS World University Rankings 2015-2016.

Tabel 1-9 Peringkat QS World University Rankings 2015-2016

Peringkat QS	Nama Perguruan Tinggi
1	Universitas Indonesia (UI)
2	Institut Teknologi Bandung (ITB)
3	Universitas Gadjah Mada (UGM)
4	Universitas Airlangga (UNAIR)
5	Institu Pertanian Bogor (IPB)
6	Universitas Padjadjaran (UNPAD)
7	Universitas Diponegoro (UNDIP)
8	Universitas Muhammadiyah Surakarta (UMS)
9	Institut Teknologi Sepuluh Nopember (ITS)
10	Universitas Brawijaya (UB)

11	Universitas Bina Nusantara (BINUS)
----	------------------------------------

Tabel 1-10 Peringkat Reputasi Akademik QS 2015-2016

Peringkat Reputasi Akademik	Nama Perguruan Tinggi
1	Institut Teknologi Bandung (ITB)
2	Universitas Indonesia (UI)
3	Universitas Gadjah Mada (UGM)
4	Universitas Muhammadiyah Surakarta (UMS)
5	Universitas Airlangga (UNAIR)
6	Institu Pertanian Bogor (IPB)
7	Universitas Diponegoro (UNDIP)
8	Universitas Brawijaya (UB)
9	Institut Teknologi Sepuluh Nopember (ITS)
10	Universitas Padjadjaran (UNPAD)
11	Universitas Islam Indonesia (UII)
12	Universitas 17 Agustus 1945 (UNTAG)
13	Universitas Bina Nusantara (BINUS)
14	Universitas Pelita Harapan (UPH)
15	Universitas Sebelas Maret (UNS)
16	Universitas Mataram (UNRAM)

2. QS 2016-2017

Peringkat universitas wilayah Indonesia berdasarkan QS World University Rankings 2016-2017.

Tabel 1-11 Peringkat QS World University Rankings 2016-2017

Peringkat QS	Nama Perguruan Tinggi
1	Universitas Indonesia (UI)
2	Institut Teknologi Bandung (ITB)

3	Universitas Gadjah Mada (UGM)
4	Universitas Airlangga (UNAIR)
5	Institu Pertanian Bogor (IPB)
6	Universitas Diponegoro (UNDIP)
7	Institut Teknologi Sepuluh Nopember (ITS)
8	Universitas Muhammadiyah Surakarta (UMS)
9	Universitas Brawijaya (UB)

Tabel 1-12 Peringkat Reputasi Akademik QS 2016-2017

Peringkat Reputasi Akademik	Nama Perguruan Tinggi
1	Institut Teknologi Bandung (ITB)
2	Universitas Indonesia (UI)
3	Universitas Gadjah Mada (UGM)
4	Universitas Airlangga (UNAIR)
5	Institu Pertanian Bogor (IPB)
6	Universitas Diponegoro (UNDIP)
7	Institut Teknologi Sepuluh Nopember (ITS)
8	Universitas Muhammadiyah Surakarta (UMS)
9	Universitas Brawijaya (UB)

3. QS 2017-2018

Peringkat universitas wilayah Indonesia berdasarkan QS World University Rankings 2017-2018.

Tabel 1-13 Peringkat QS World University Rankings 2017-2018

Peringkat QS	Nama Perguruan Tinggi
1	Universitas Indonesia (UI)
2	Institut Teknologi Bandung (ITB)

3	Universitas Gadjah Mada (UGM)
4	Universitas Airlangga (UNAIR)
5	Institu Pertanian Bogor (IPB)
6	Universitas Diponegoro (UNDIP)
7	Institut Teknologi Sepuluh Nopember (ITS)
8	Universitas Muhammadiyah Surakarta (UMS)
9	Universitas Brawijaya (UB)

Tabel 1-14 Peringkat Reputasi Akademik QS 2017-2018

Peringkat Reputasi Akademik	Nama Perguruan Tinggi
1	Institut Teknologi Bandung (ITB)
2	Universitas Gadjah Mada (UGM)
3	Universitas Indonesia (UI)
4	Universitas Airlangga (UNAIR)
5	Institu Pertanian Bogor (IPB)
6	Universitas Diponegoro (UNDIP)
7	Institut Teknologi Sepuluh Nopember (ITS)
8	Universitas Muhammadiyah Surakarta (UMS)
9	Universitas Brawijaya (UB)

3.1.9.2 Analisa Label Topik Dokumen

Penggunaan teks mining menggunakan metode LDA-AdaBoost.MH membutuhkan acuan kebenaran sebagai perbandingan apakah keluaran sistem yang dibangun sudah sesuai dengan kebenaran dalam melakukan penentuan topik dalam indikator reputasi akademik.

Ground Truth yang digunakan adalah hasil analisa manual terhadap dokumen abstrak *paper* akademisi perguruan tinggi.

Terdapat 800 dokumen sebagai dataset metode teks mining dari masing-masing perguruan tinggi, dan dianalisa manual untuk mencari label topik apa yang sesuai. Berikut adalah beberapa contoh hasil analisa manual terhadap dataset.

Tabel 1-15 Contoh Hasil Analisa Manual Sebagai Ground Truth Reputasi Akademik

No	Nama Perguruan Tinggi	No Dok	Nama Dokumen	Hasil Analisa
1	Institut Pertanian Bogor	1	IPB_001_01	T1, T2
		2	IPB_001_02	T1, T2
		3	IPB_001_03	T1, T2, T3
		4	IPB_05_10	T3, T4, T5
		5	IPB_05_14	T4, T5, T6, T7
2	Institut Teknologi Bandung	1	ITB_01_01	T4, T3, T5
		2	ITB_01_05	T4, T5
		3	ITB_01_22	T1, T2, T3
		4	ITB_01_09	T4, T5, T6, T7
		5	ITB_01_07	T4, T5

3.1.10 Dataset Pengujian Sistem

3.1.10.1 Data Dokumen Abstrak Akademisi Perguruan Tinggi

Dokumen abstrak paper akademisi didapatkan dari database metadata publikasi Google Scholar. Dalam Tugas Akhir ini memfokuskan dokumen abstrak paper pada akademisi yang mempunyai bidang fokus teknik. Masing-masing perguruan tinggi diambil 50 dokumen abstrak *paper* teknik berdasarkan sitasi tertinggi dari akademisi yang terdaftar atas institusi perguruan tinggi di Google Scholar.

Dalam proses teks mining, 50 buah dataset abstrak paper perguruan tinggi digabung menjadi satu dokumen utuh yang memungkinkan adanya kata yang sama berjumlah lebih dari satu (duplikasi). Satu dokumen utuh ini selanjutnya akan menjadi

masukkan untuk proses tek mining tahap preprocessing. Penentuan dataset dokumen abstrak paper akademisi mempunyai rule seperti berikut:

1. Pencarian daftar akademisi dalam Google Scholar dengan keyword email yang terhubung dengan institusi atau keyword hosting institusi. Misal Institut Teknologi Sepuluh Nopember dicari menggunakan keyword “@its.ac.id” atau www.its.ac.id
2. Daftar akademisi yang diambil adalah yang mempunyai tingkat sitasi tertinggi dan berfokus pada bidang teknik.
3. Masing-masing dari akademisi yang mempunyai sitasi tertinggi diambil paper yang membahas teknik.
4. Dokumen abstrak paper difokuskan pada dokumen berbahasa Inggris.

Tahap proses teks mining dalam Tugas Akhir ini akan menghasilkan reduksi terhadap kata (*term*) yang ada di dalam dokumen abstrak *paper* akademisi. Reduksi terhadap kata (*term*) berperan penting untuk menyeleksi kata (*term*) yang tidak penting. Dilakukan analisa terhadap keberadaan kata (*term*) yang penting dan tidak penting dalam dataset untuk melihat rata-rata duplikasi term. Rincian jumlah kata pada dataset abstrak paper akademisi (16 perguruan tinggi teratas berdasarkan *Ground Truth*) seperti pada Tabel 1-16.

Tabel 1-16 Daftar Rincian Jumlah Kata Dataset Abstrak Paper Perguruan Tinggi

No.	Perguruan Tinggi	Total Kata
1.	Institus Pertanian Bogor	10539
2.	Institut Teknologi Bandung	9287
3.	Institut Teknologi Sepuluh Nopember	9630
4.	Universitas 17 Agustus 1945	9146
5.	Universitas Airlangga	9457
6.	Universitas Bina Nusantara	7236
7.	Universitas Brawijaya	8651
8.	Universitas Diponegoro	8397
9.	Universitas Gadjah Mada	10214

10.	Universitas Indonesia	8823
11.	Universitas Islam Indonesia	7897
12.	Universitas Mataram	7527
13.	Universitas Muhammadiyah Surakarta	10330
14.	Universitas Padjadjaran	7774
15.	Universitas Pelita Harapan	7204
16.	Universitas Sebelas Maret	7807

3.1.10.2 Data Corpus Taxonomy Bloom

Corpus Taxonomy Bloom berfungsi sebagai dasar pengetahuan corpus tingkat kesiapterapan teknologi (TKT). Corpus Taxonomy Bloom mempunyai 6 level tingkatan yaitu Knowledge, Comprehension, Application, Analysis, Synthesis, dan Evaluation (Center of Teaching and Learning Stanford University, 1981). 6 level corpus Taxonomy Bloom akan digunakan untuk pengetahuan corpus TKT yang nantinya akan dikembangkan lagi menjadi 9 level (*class*). Dilihat dari jumlah keyword yang tercakup dalam masing-masing level Taxonomy Bloom dapat dikatakan kurang banyak dalam jumlah keyword jika dibandingkan dengan jumlah kata (*term*) pada dokumen dataset abstrak *paper* akademisi. Untuk selanjutnya akan diadakan proses breakdown corpus Taxonomy Bloom menjadi corpus TKT dengan melalui berbagai tahap. Rincian jumlah dari masing-masing corpus Taxonomy Bloom seperti pada Tabel 1-17.

Tabel 1-17 Daftar Rincian Jumlah Kata Corpus Taxonomy Bloom

No.	Level Taxonomy	Total Kata
1.	Knowledge	35
2.	Comprehension	29
3.	Application	36
4.	Analysis	51
5.	Synthesis	51
6.	Evaluation	46

3.1.10.3 Data Corpus Tingkat Kesiapterapan Teknologi (TKT)

Corpus tingkat kesiapterapan teknologi (TKT) merupakan corpus utama untuk proses pengenalan kata (*term*) dalam dokumen abstrak *paper* akademisi agar masuk dalam kategori level TKT yang mana. Corpus TKT ini berjumlah 9 level, dan didapatkan dari proses breakdown 6 level corpus Taxonomy Bloom.

Metode breakdown yang digunakan dalam penentuan corpus TKT ini adalah sebagai berikut:

1. Menggabungkan 6 level corpus Taxonomy Bloom menjadi satu dokumen besar.
2. Mengurutkan kata (*term*) dari corpus Taxonomy Bloom besar berdasarkan bobot konteks kata (*term*) dari yang paling dasar hingga tingkat penerapan. Tahap ini menggunakan kemampuan manusia (*expert*) untuk membedakan dan mensorting sesuai urutan.
3. Menghapus kata yang berjumlah lebih dari satu (duplikasi).
4. Memecah urutan konten kata (*term*) tersebut menjadi 9 bagian. Pemecahan ini didasarkan pada setiap urutan level TKT akan bergantung pada TKT selanjutnya. Sebagai contoh corpus TKT 2 pasti berkorelasi dengan TKT 1 dan TKT 3. Sehingga pemecahan ini mudah untuk dilakukan. Secara tidak langsung level TKT dapat diperoleh kata-kata dari berbagai macam kata pada level Taxonomy Bloom.
5. Dokumen hasil breakdown berjumlah 9 corpus kata TKT baru tanpa mengandung kata duplikasi.

Perbandingan dari 6 level corpus Taxonomy Bloom dengan 9 level corpus TKT seperti pada Tabel 1-18.

Tabel 1-18 Perbandingan Jumlah Kata Corpus TKT

Level Taxonomy Bloom	Jumlah Kata	Level TKT	Jumlah Kata
Knowledge	35	TKT 1	31
		TKT 2	24
Comprehension	29	TKT 3	32

Application	36	TKT 4	15
		TKT 5	32
Analysis	51	TKT 6	24
Synthesis	51	TKT 7	26
		TKT 8	30
Evaluation	46	TKT 9	34
Total	256	Total	256

3.2 Tahap Perancangan

Tahap ini meliputi perancangan basis data, perancangan tampilan antarmuka (*Interfaces*), perancangan alur proses penggunaan sistem antar pengguna, dan proses utama dalam proses teks mining topik modelling menggunakan metode LDA-AdaBoost.MH. Diharapkan dapat memenuhi tujuan dari pengembangan aplikasi ini.

3.2.1 Perancangan Basis Data

Perangkat lunak yang dibangun menggunakan basis data MySQL dengan bahasa syntax SQL. Terdapat 28 tabel untuk menampung hasil proses grabbing data online, hasil proses teks minning berupa skor reputasi akademik, hasil penampung skor masing-masing kriteria, serta data perguruan tinggi. Terdapat 2 buah kegunaan utama dari basis data yang digunakan dalam pembuatan perangkat lunak, yaitu sebagai penyimpan data final, dan penyimpan data temporary update. Dengan rincian 16 tabel utama dan 12 tabel temporary sebagai penampung sementara data hasil Teknik grabbing.

3.2.1.1 Tabel Universitas

Identifikasi : universitas

Deskripsi isi : tabel untuk menyimpan data nama perguruan tinggi beserta informasi detail

Id_field	Deskripsi	Tipe & length	Null
----------	-----------	---------------	------

Id	Variabel primary key	Int(2)	No
Nama	Variabel penampung nama	Varchar(255)	No
Domain	Variabel penampung domain webhost	Varchar(150)	No
url_web	Variabel penampung alamat link webhost	Varchar(150)	No
Kota	Variabel kota	Varchar(150)	No
provinsi	Variabel provinsi	Varchar(150)	No

3.2.1.2 Tabel Reputasi Akademik

Identifikasi : reputasi_akademik

Deskripsi isi : tabel untuk menyimpan nilai hasil proses teks mining metode LDA-AdaBoost.MH sebagai representasi nilai reputasi akademik

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Nama_universitas	Variabel berisi nama perguruan tinggi	Varchar(255)	No
Skor_universitas	Variabel penampung skor final reputasi akademik	Float	No
Bobot_total	Variabel penampung jumlah hasil reputasi akademik	Int(2)	No
Jumlah_dokumen	Variabel penampung jumlah dokumen olah	Int(2)	No
Tanggal_akses	Variabel timestamp	Datetime	No

3.2.1.3 Tabel Akreditasi

Identifikasi : tampung_akreditasi

Deskripsi isi : tabel untuk menyimpan nilai hasil proses grabbing akreditasi perguruan tinggi

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Nama_universitas	Variabel berisi nama perguruan tinggi	Varchar(255)	NO

Skor_universitas	Variabel penampung skor final reputasi akademik	Float	NO
Bobot_total	Variabel penampung jumlah hasil reputasi akademik	Int(2)	No
Jumlah_dokumen	Variabel penampung jumlah dokumen olah	Int(2)	No
Tanggal_akses	Varabel waktu akses	Datetime	No

3.2.1.4 Tabel Jumlah Mahasiswa

Identifikasi : jumlah_mahasiswa

Deskripsi isi : tabel untuk menyimpan jumlah mahasiswa yang didapatkan dari data PDDIKTI

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Kode_universitas	Variabel penampung kode status diakui universitas	Int(2)	No
Nama_universitas	Variabel berisi nama perguruan tinggi	Varchar(255)	No
Jumlah_tahun1_semester1	Variabel jumlah mahasiswa tahun1 pada semester1	Int(2)	No
Jumlah_tahun1_semester2	Variabel jumlah mahasiswa tahun1 pada semester2	Int(2)	No
Jumlah_tahun2_semester1	Variabel jumlah mahasiswa tahun2 pada semester1	Int(2)	No
Jumlah_tahun2_semester2	Variabel jumlah mahasiswa tahun2 pada semester2	Int(2)	No
Status_universitas	Variabel berisi keterangan status universitas	Varchar(5)	No

3.2.1.5 Tabel Jumlah Sitasi Sinta

Identifikasi : jumlah_sitasi_sinta

Deskripsi isi : tabel untuk menyimpan jumlah sitasi akademisi yang didapatkan dari database online SINTA

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Kode_universitas	Variabel penampung kode status diakui universitas	Int(2)	No
Nama_universitas	Variabel berisi nama perguruan tinggi	Varchar(250)	No
Jumlah_tahun1_semester1	Variabel jumlah mahasiswa tahun1 pada semester1	Int(2)	No
Jumlah_tahun1_semester2	Variabel jumlah mahasiswa tahun1 pada semester2	Int(2)	No
Jumlah_tahun2_semester1	Variabel jumlah mahasiswa tahun2 pada semester1	Int(2)	No
Jumlah_tahun2_semester2	Variabel jumlah mahasiswa tahun2 pada semester2	Int(2)	No
Status_universitas	Variabel berisi keterangan status universitas	Varchar(5)	No

3.2.1.6 Tabel Jumlah Author Sinta

Identifikasi : jumlah_author_sinta

Deskripsi isi : tabel untuk menyimpan jumlah h-index dan i10-index akademisi yang didapatkan dari database online SINTA

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Id_data	Variabel tampung id data secara kolektif	Int(2)	No
Nomor	Variabel nomor akademisi dalam range satu kali iterasi	Int(2)	No
Nama_author	Variabel penampung nama akademisi	Varchar(250)	No
Nama_universitas	Variabel penampung nama universitas	Varchar(250)	No

Jumlah_sitasi_scholar	Variabel jumlah sitasi	Int(2)	No
I10_index_scholar	Vaiabel jumlah i10-index	Int(2)	No
H_index_scholar	Variabel jumlah h_index	Int(2)	No
Waktu_akses	Variabel penampung waktu akses	Tinytext	No

3.2.1.7 Tabel Precense

Identifikasi : precense

Deskripsi isi : tabel untuk menyimpan jumlah domain, subdomain dan repository webhost perguruan tinggi yang terindex Google

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
url_web	Variabel penampung link webhost	Varchar(255)	No
Domain_search	Variabel penampung link search webhost	Varchar(150)	No
Hasil_search	Variabel penampung hasil search webhost	Int(2)	No
Tanggal_akses	Variabel waktu akses	Datetime	No

3.2.1.8 Tabel Impact

Identifikasi : impact

Deskripsi isi : tabel untuk menyimpan jumlah page authority, domain authority dan backlink webhost perguruan tinggi

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
url_web	Variabel penampung link webhost	Varchar(255)	No
Domain_web	Variabel penampung link webhost tanpa “www”	Varchar(150)	No

Page_auth ority	Variabel penampung nilai page authority	Int(2)	No
Domain_a uthority	Variabel penampung nilai domain authority	Int(2)	No
Backlink_e ksterna	Variabel penampung jumlah backlink	Int(2)	No
Tanggal_a kses	Variabel waktu akses	Datetime	No

3.2.1.9 Tabel Alexa Rank

Identifikasi : alexa_rank

Deskripsi isi : tabel untuk menyimpan peringkat webhost
berdasarkan alexa rank

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
url_web	Variabel penampung link webhost	Varchar(255)	No
Rank_worl d	Variabel penampung peringkat webhost skala dunia	Int(2)	No
Rank_coun try	Variabel penampung peringkat webhost skala nasional	Int(2)	No
Country_n ame	Variabel penampung nilai domain authority	Varchar(150)	No
Tanggal_a kses	Variabel waktu akses	Datetime	No

3.2.1.10 Tabel Kumpulan Skor Final

Identifikasi : kumpulan_skor_final

Deskripsi isi : tabel untuk menyimpan skor dari setiap kriteria
pemeringkatan

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Kode_univer sitas	Variabel penampung kode status diakui universitas	Float	No
Nama_unive rsitas	Variabel nama perguruan tinggi	Varchar(255)	No

Skor_reputasi_akademik_normalisasi	Variabel penampung skor reputasi akademik yang dinormalisasi	Float	No
Skor_nilai_akreditasi_normalisasi	Variabel penampung skor nilai akreditasi yang dinormalisasi	Float	No
Skor_jumlah_mahasiswa_normalisasi	Valriabel penampung skor jumlah mahasiswa yang dinormalisasi	Float	No
Skor_jumlah_sitasi_normalisasi	Variabel penampung skor jumlah sitasi yang dinormalisasi	Float	No
Skor_jumlah_tenaga_pengajar_normalisasi	Variabel penampung skor jumlah tenaga pengajar yang dinormalisasi	Float	No
Skor_kualitas_sitasi_normalisasi	Variabel penampung skor kualitas sitasi yang sudah dinormalisasi	Float	No
Skor_jumlah_phd_normalisasi	Variabel penampung skor jumlah tenaga pengajar S3 (Phd) yang sudah dinormalisasi	Float	No
Skor_jumlah_paper_normalisasi	Variabel penampung skor jumlah paper yang dinormalisasi	Float	No
Skor_jumlah_jurnal_artikel	Variabel penampung skor jumlah jurna artikel yang dinormalisasi	Float	No
Skor_jumlah_paper_conference	Variabel penampung skor jumlah paper conference yang dinormalisasi	Float	No
Skor_precense_normalisasi	Variabel penampung skor precense yang dinormalisasi	Float	No
Skor_impact_normalisasi	Variabel penampung skor impact yang dinormalisasi	Float	No

Skor_transpa rency_norma lisasi	Variabel penampung skor transparency yang dinormalisasi	Float	No
Skor_exellec en_normalis asi	Variabel penampung skor excellence yang dinormalisasi	Float	No
Skor_moz_a nalytic_nor malisasi	Variabel penampung skor moz analytic yang dinormalisasi	Float	No
Skor_alexa_ rank_normal isasi	Variabel penampung skor alexa rank yang dinormalisasi	Float	No
Skor_kualita s_dosen_nor malisasi	Variabel penampung skor kualitas dosen yang dinormalisasi	Float	No
Skor_kecuku pan_dosen_ normalisasi	Variabel penampung skor kecukupan dosen yang dinormalisasi	Float	No
Skor_kualita s_manajeme n_normalisa si	Variabel penampung skor kualitas manajemen yang dinormalisasi	Float	No
Skor_kualita s_kegiatan_k emahasiswaa n	Variabel penampung skor kualitas kegiatan mahasiswa yang dinormalisasi	Float	No
Sko_kualitas _kegiatan_p enelitian	Variabel penampung skor kualitas kegiatan penelitian yang dinormalisasi	Float	No

3.2.1.11 Tabel Peringkat 4ICU

Identifikasi : 4icu

Deskripsi isi : tabel untuk menyimpan skor dan peringkat pemeringkatan 4ICU

<u>Id_field</u>	<u>Deskripsi</u>	<u>Tipe & length</u>	<u>Null</u>
Id	Variabel primary key	Int(2)	No
Nama_unive rsitas	Variabel penampung nama universitas	Varchar(255)	No

Skor_4icu	Variabel penampung skor 4ICU	Float	No
Peringkat_4icu	Variabel penampung peringkat 4ICU	Int(2)	No

3.2.1.12 Tabel Peringkat DIKTI

Identifikasi : dikti

Deskripsi isi : tabel untuk menyimpan skor dan peringkat pemerinkkatan DIKTI Indonesia

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Nama_universitas	Variabel penampung nama universitas	Varchar(150)	No
Skor_dikti	Variabel penampung skor DIKTI	Float	No
Peringkat_dikti	Variabel penampung peringkat DIKTI	Int(2)	No

3.2.1.13 Tabel Peringkat QS General

Identifikasi : qs_general

Deskripsi isi : tabel untuk menyimpan skor dan peringkat pemerinkkatan QS rank versi general

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Nama_universitas	Variabel penampung nama universitas	Varchar(255)	No
Skor_qs	Variabel penampung skor QS General	Float	No
Peringkat_qs	Variabel penampung peringkat QS General	Int(2)	No

3.2.1.14 Tabel Peringkat QS Asia

Identifikasi : qs_asia

Deskripsi isi : tabel untuk menyimpan skor dan peringkat pemerinkkatan QS rank versi region Asia

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Nama_universitas	Variabel penampung nama universitas	Varchar(255)	No
Skor_qs_asia	Variabel penampung skor QS Asia	Float	No
Peringkat_qs-asia	Variabel penampung peringkat QS Asia	Int(2)	No

3.2.1.15 Tabel Peringkat QS Metode LDA-AdaBoost.MH

Identifikasi : qs_llda

Deskripsi isi : tabel untuk menyimpan skor dan peringkat pemeringkatan QS rank berdasarkan metode LDA-AdaBoost.MH

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Nama_universitas	Variabel penampung nama universitas	Varchar(255)	No
Skor_llda	Variabel penampung skor QS LDA-AdaBoost.MH	Float	No
Peringkat_llda	Variabel penampung peringkat QS LDA-AdaBoost.MH	Int(2)	No

3.2.1.16 Tabel Peringkat Webometrics

Identifikasi : webometrics

Deskripsi isi : tabel untuk menyimpan skor dan peringkat pemeringkatan Webometrics

Id_field	Deskripsi	Tipe & length	Null
Id	Variabel primary key	Int(2)	No
Nama_universitas	Variabel penampung nama universitas	Varchar(255)	No
Skor_webometrics	Variabel penampung skor Webometrics	Float	No
Peringkat_webometrics	Variabel penampung peringkat Webometrics	Int(2)	No

3.2.2 Perancangan Antarmuka

Perangkat lunak yang dibangun berbasis pada aplikasi website. Antarmuka perangkat lunak dibangun menggunakan bahasa pemrograman PHP dan HTML. Berikut adalah rincian antarmuka yang dibangun dalam perangkat lunak.

3.2.2.1 Antarmuka Home

Identifikasi : antarmuka home (index)

Deskripsi isi : antarmuka ini berfungsi sebagai halaman root utama dalam perangkat lunak website. Terdapat satu kebutuhan fungsional dalam antarmuka ini yaitu pencarian perguruan tinggi berdasarkan potongan nama. Terdapat kolom search perguruan tinggi.

3.2.2.2 Antarmuka Latar Belakang Pemeringkatan

Identifikasi : antarmuka sekilas pemeringkatan

Deskripsi isi : antarmuka ini berfungsi sebagai penampil informasi sederhana yang menyatakan latar belakang kegunaan dari sistem pemeringkatan secara general.

3.2.2.3 Antarmuka Metode Penilaian

Identifikasi : antarmuka metode penilaian

Deskripsi isi : antarmuka ini berfungsi sebagai penampil informasi mengenai kriteria penilaian apa saja yang digunakan dalam sistem pemeringkatan. Terdapat informasi secara general dan informasi secara detail dari setiap tipe pemeringkatan, mulai dari pemeringkatan QS General, QS Asia, Webometrics, 4ICU, pemeringkatan DIKTI, dan QS menggunakan metode LDA-AdaBoost.MH. Terdapat tombol untuk melihat tipe pemeringkatan yang diinginkan.

3.2.2.4 Antarmuka Peringkat Berdasarkan Region

Identifikasi : antarmuka peringkat berdasarkan region

Deskripsi isi : antarmuka ini berfungsi sebagai halaman pencarian peringkat perguruan tinggi berdasarkan region wilayah provinsi perguruan tinggi. Dengan menampilkan desain map negara Indonesia sebagai inputan pencarian yang mudah untuk dipahami. Pencarian dilakukan dengan cara menekan salah satu bagian provinsi dari map negara Indonesia.

3.2.2.5 Antarmuka Hasil Pencarian Berdasar Region Provinsi

Identifikasi : antarmuka hasil pencarian berdasarkan region provinsi

Deskripsi isi : antarmuka ini berfungsi sebagai halaman penampil hasil pencarian perguruan tinggi berdasarkan region provinsi yang dimiliki. Ditampilkan juga peringkat dari perguruan tinggi dalam hasil pencarian dari beberapa tipe pemeringkatan seperti QS General, QS Asia, Webometrics, 4ICU, pemeringkatan DIKTI, dan QS LDA-AdaBoost.MH. Terdapat tombol untuk melihat tipe pemeringkatan yang diinginkan, dan terdapat kolom filter untuk mencari berdasarkan kota.

3.2.2.6 Antarmuka Hasil Pencarian Berdasar Pencarian Keyword Potongan Nama

Identifikasi : antarmuka hasil pencarian berdasarkan pencarian keyword potongan nama

Deskripsi isi : antarmuka ini berfungsi sebagai halaman penampil hasil pencarian perguruan tinggi berdasarkan keyword potongan nama. Terdapat informasi peringkat dari masing-masing perguruan tinggi yang dihasilkan. Terdapat tombol “Lihat Detail” untuk melihat detail informasi akademik perguruan tinggi.

3.2.2.7 Antarmuka Detail Informasi Akademik

Identifikasi : antarmuka detail informasi akademik

Deskripsi isi : antarmuka ini berfungsi sebagai halaman penampil informasi akademik perguruan tinggi. Terdapat beberapa informasi diantaranya mahasiswa, kecukupan dosen, status akreditasi, kualitas akademisi, produktifitas institusi, dan kepopuleran website. Ditampilkan juga grafik peringkat dari berbagai tipe pemeringkatan.

3.2.2.8 Antarmuka Pembobotan Pemeringkatan

Identifikasi : antarmuka pembobotan pemeringkatan

Deskripsi isi : antarmuka ini berfungsi sebagai halaman masukan bobot pemeringkatan yang diinginkan. Terdapat tabel tipe pemeringkatan dan mempunyai daftar kriteria yang digunakan. Pengguna dapat memasukkan bobot pemeringkatan sesuai kebutuhan. Nilai maksimal pembobotan dari satu tipe pemeringkatan adalah 100%, jika lebih dari itu maka akan muncul pesan error bobot berlebihan. Terdapat tombol pencarian berdasarkan tipe pemeringkatan, diantaranya “Cek QS-General”, “Cek QS-Asia”, “Cek 4ICU”, “Cek Webometrics”, “Cek DIKTI”, “Cek QS L-LDA”, “Cek QS PLSa”, “Cek QS Helmholtz”, “Cek QS Adaboost”.

3.2.2.9 Antarmuka Hasil Pemeringkatan Berdasar Pembobotan Tipe Pemeringkatan yang Diinginkan

Identifikasi : antarmuka hasil pemeringkatan berdasarkan hasil pembobotan

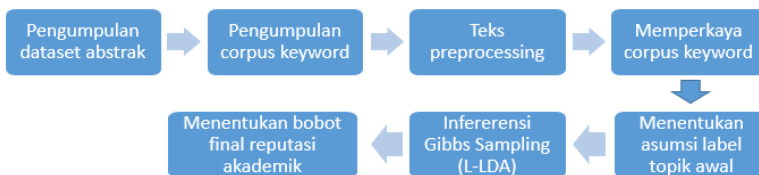
Deskripsi isi : antarmuka ini berfungsi sebagai halaman penampil hasil pembobotan perguruan tinggi sesuai dengan tipe pemeringkatan yang dikehendaki. Terdapat kolom filter untuk mencari berdasarkan kota.

3.2.3 Perancangan Kriteria Penilaian Perguruan Tinggi

Dalam menentukan peringkat perguruan tinggi menggunakan metode LDA-AdaBoost.MH, dibutuhkan beberapa kriteria penilaian lain sebagai penunjang aspek kriteria kuantitatif dan kualitatif (QS World University Rankings, 2016). Penjelasan masing-masing kriteria akan dijelaskan dibawah ini.

3.2.3.1 Reputasi Akademik

Merupakan kriteria utama yang bersifat kualitatif yang didapatkan dari proses teks mining dokumen abstrak perguruan tinggi menggunakan metode LDA-AdaBoost.MH (Daniel Ramage, David Hall, Ramesh Nallapati, Christopher D. Manning, 2009). Mempunyai bobot pemeringkatan 40% dari total bobot keseluruhan. Alur proses menghasilkan kriteria reputasi akademik seperti pada Gambar 3.2-1.



Gambar 3.2-1 Alur Menentukan Reputasi Akademik

Langkah pertama dengan mengumpulkan dataset berupa abstrak paper perguruan tinggi, kemudian membangun corpus keyword yang didapatkan dari taxonomy Bloom dan dikembangkan menjadi corpus TKT seperti pada langkah 4 pada bagan diatas. Selanjutnya melakukan tahap teks preprocessing untuk menghasilkan kata inputan yang baku dari dokumen dataset. Hasil dari tahap teks preprocessing akan digunakan sebagai penentuan asumsi label topik awal, dimana setiap dokumen harus mempunyai label sebagai acuan topiknya. Kemudian masuk pada tahap inferensi Gibbs Sampling untuk menghasilkan probabilitas terbaik dari dokumen terhadap topik. Hasil dari tahap ini berupa

probabilitas akhir yang nantinya akan dihitung menggunakan pembobotan final reputasi akademik. Hasil ini kemudian disimpan dalam database sebagai nilai indicator reputasi akademik.

3.2.3.2 Akreditasi Perguruan Tinggi

Merupakan kriteria yang bersifat kuantitatif. Didapatkan dari proses grabbing data online pada website BAN-PT. Data yang diambil adalah data nama perguruan tinggi, masa berlaku, nilai huruf akreditasi, serta status perguruan tinggi. Mempunyai bobot 10% dari total bobot keseluruhan. Alur proses menghasilkan kriteria akreditasi perguruan tinggi seperti pada Gambar 3.2-2.



Gambar 3.2-2 Alur Menentukan Skor Akreditasi

Langkah pertama dengan menyiapkan situs website yang akan dijadikan sasaran *grabbing*, dalam Tugas Akhir ini menggunakan 2 percobaan pada website BAN-PT dan website penyedia informasi akreditasi. Kemudian Menentukan teknik grabbing apa dan library yang akan digunakan, dalam Tugas Akhir ini menggunakan Bahasa pemrograman PHP, library HTML DOM Parser, dan beberapa fungsi bawaan dari PHP sendiri. Kemudian melakukan grabbing dengan ketentuan harus mempunyai bandwidth yang cukup. Karena proses grabbing membutuhkan waktu yang cukup lama melihat konten yang ingin diambil cukup besar dan banyak. Dalam proses grabbing data yang masuk akan langsung disimpan dalam table temporary pada database, sebagai penampung sementara karena tujuan utama disini adalah mengambil sebisa mungkin. Dalam database table temporary ini kemudian akan dianalisa dan dilakukan normalisasi hasil untuk menghasilkan skor akreditasi yang paling akhir. Skor akreditasi ini kemudian akan disimpan lagi dalam database dengan table yang

berbeda sebagai implementasi skor final akreditasi perguruan tinggi.

3.2.3.3 Jumlah Rasio Mahasiswa dan Tenaga Pengajar Aktif

Merupakan kriteria yang bersifat kuantitatif. Didapatkan dari data rekap database online PDDIKTI. Data yang diambil merupakan rekap yang berisi jumlah mahasiswa, nama perguruan tinggi serta jumlah tenaga pengajar keseluruhan. Mempunyai bobot 20% dari total bobot keseluruhan. Alur proses menghasilkan kriteria rasio mahasiswa dan tenaga pengajar aktif seperti pada Gambar 3.2-3.



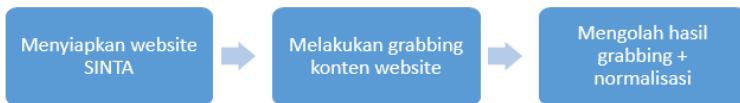
Gambar 3.2-3 Alur Menentukan Skor Rasio Mahasiswa dan Tenaga Penagajar Aktif

Langkah pertama dengan mengumpulkan data rakap jumlah mahasiswa dan rekap jumlah tenaga pengajar dari seluruh perguruan tinggi di Indonesia. Data ini disediakan oleh PDDIKTI sebagai pusat databasenya. Untuk mempermudah proses analisa, dalam Tugas Akhir ini menggunakan data rekap berformat .xlsx kemudian akan dibaca menggunakan library excel dalam Bahasa pemrograman python. Karena file excel berupa row dan column maka akan diambil beberapa variable penting sebagai acuan penilaian. Variable jumlah mahasiswa semester 1 tahun 1, mahasiswa semester 2 tahun 1, mahasiswa semester 1 tahun 2, dan semester 2 tahun 2 menjadi acuan penilaian mahasiswa. Serta data jumlah teanga pengajar secara keseluruhan dari setiap perguruan tinggi akan menjdi acuan penilaian jumlah tenaga pangajar. Nilai dari variable ini akan dinormalisasi terlebih dahulu dengan membagi dengan nilai tertinggi. Kemudian akan dirasiokan dengan jumlah tenaga pengajar, kemudian disimpan dalam database

sebagai implementasi skor final rasio mahasiswa dan tenaga pengajar aktif.

3.2.3.4 Jumlah Sitasi Akademisi Penelitian

Merupakan kriteria yang bersifat kuantitatif. Didapatkan dari proses grabbing data online Science and Technology Index (SINTA). Data yang digunakan adalah nama perguruan tinggi, serta jumlah sitasi dan author akademisi yang terindex SINTA. Mempunyai bobot 20% dari total bobot keseluruhan. Alur proses menghasilkan kriteria jumlah sitasi akademisi penelitian seperti pada Gambar 3.2-4.



Gambar 3.2-4 Alur Menentukan Skor Jumlah Sitasi Akademisi Penelitian

Langkah pertama dengan menyiapkan website sasaran grabbing yaitu situs website Science and Technology Index (SINTA) dari Kemenristek Dikti. Dalam situs ini terdapat beberapa variable penting sebagai acuan penilaian indikator. Setiap variable akan dilakukan teknik *grabbing* dan sama seperti pada proses menentukan indikator akreditasi perguruan tinggi. Dalam proses ini juga membutuhkan database table temporary sebagai penampung sementara. Variable yang digunakan adalah jumlah sitasi dari google scholar dan scopus. Kemudian dari table temporary ini akan dianalisa dan dinormalisasi dan dicari rata-ratanya sebagai skor final jumlah sitasi akademisi penelitian dan dimasukkan lagi dalam table final.

3.2.3.5 Jumlah Tenaga Pengajar (Dosen)

Merupakan kriteria yang bersifat kuantitatif yang didapatkan dari data rekap database online PDDIKTI. Data yang

diambil merupakan jumlah tenaga pengajar dengan berbagai status ketenaga kerjaan, seperti pegawai tetap maupun tidak tetap. Mempunyai bobot 5% dari total bobot keseluruhan. Alur proses menghasilkan kriteria jumlah tenaga pengajar (dosen) seperti pada Gambar 3.2-3 diatas.

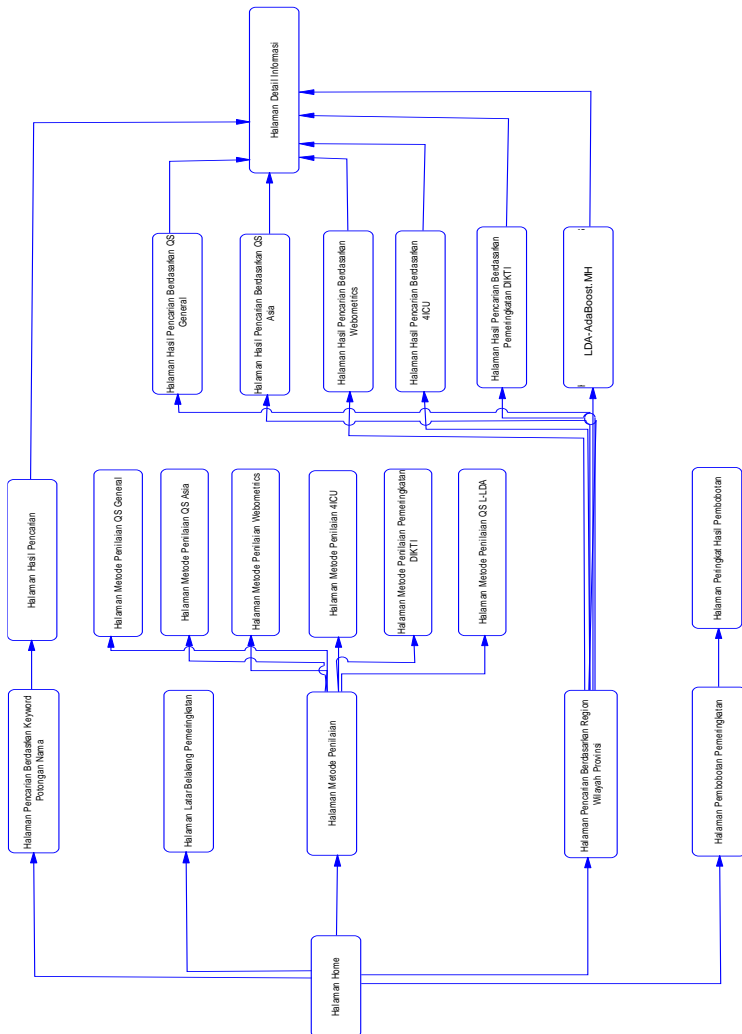
Prinsip kerja Menentukan indikator ini sama seperti indikator jumlah rasio mahasiswa dan tenaga pengajar aktif. Hanya saja disini ada beberapa data rekap yang digunakan sebagai pendukung yaitu rekap jumlah tenaga pengajar yang bersifat tetap dan tidak tetap (honorer). Data ini akan dibaca menggunakan library excel pada Bahasa pemrograman PHP kemudian akan dianalisa dan dinormalisasi untuk menghasilkan jangkauan nilai yang rata. Terdapat variable tanga pengajar tetap dan tidak tetap yang nilai masing-masing telah dinormalisasi, kemudian dijumlahkan dan dirata-rata. Nilai rata-rata ini akan menjadi skor indikator jumlah tenaga pengajar (dosen).

3.2.3.6 Kualitas Sitasi

Merupakan kriteria yang bersifat kuantitatif yang didapatkan dari proses grabbing data online Science and Technology Index (SINTA). Data yang diambil merupakan nilai h-index dan i10-index Google Scholar dari akademisi perguruan tinggi. Mempunyai bobot 5% dari total bobot keseluruhan. Alur proses menghasilkan kriteria kualitas sitasi seperti pada Gambar 3.2-4 diatas.

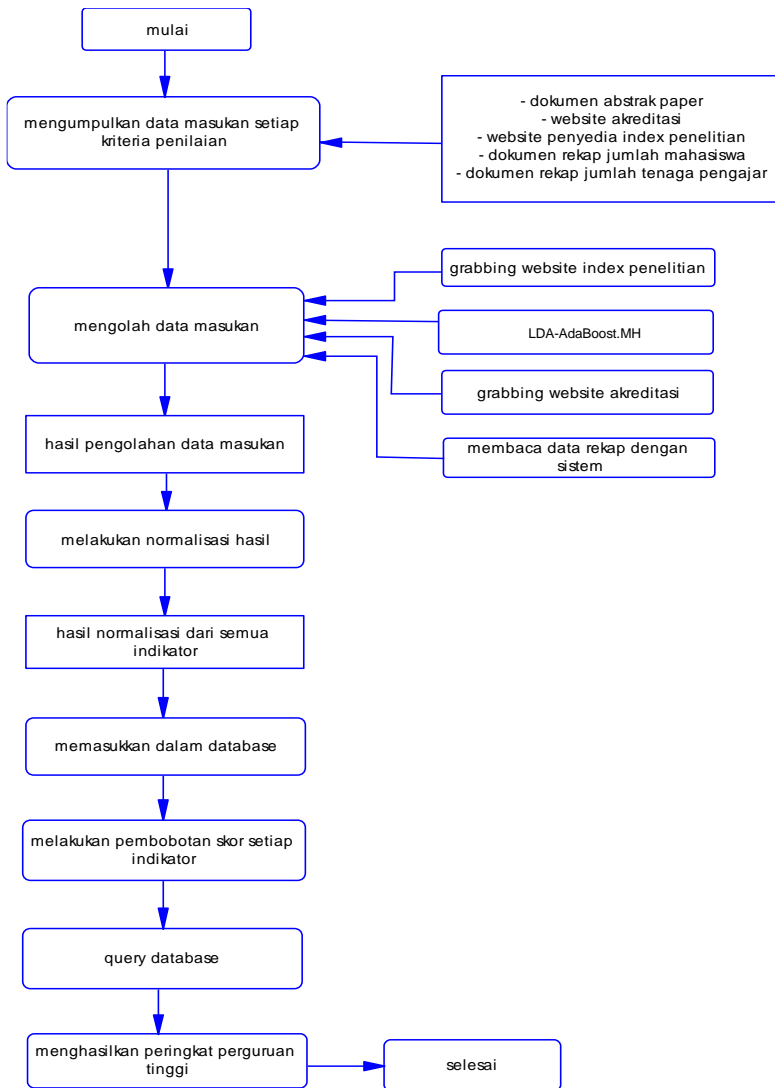
Prinsip kerja dalam Menentukan indikator ini sama seperti pada indikator jumlah sitasi akademisi penelitian. Hanya saja disini yang diambil adalah variable jumlah h-index dan i10-index dari google scholar. Kedua variable ini akan dilakukan normalisasi nilai dan dicari rata-ratanya. Dan kemudian disimpan dalam table database baru sebagai implementasi skor indikator kualitas sitasi.

3.2.4 Proses Sistem Aplikasi Pengguna (Diagram Aplikasi)



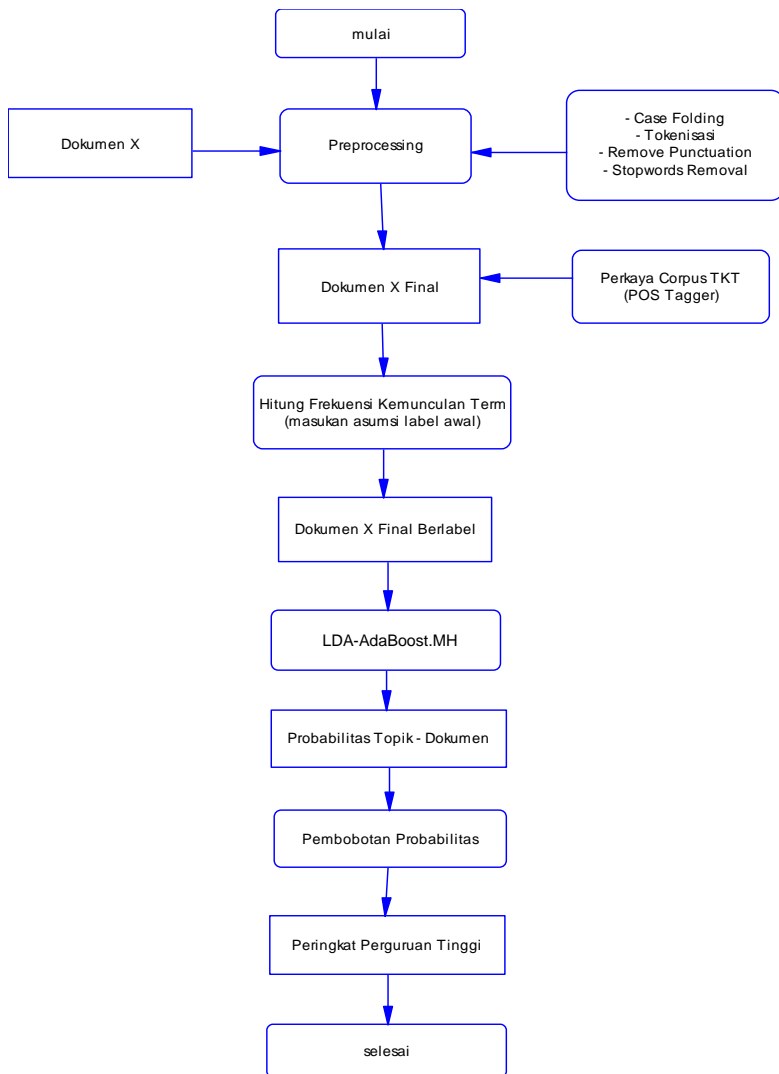
Gambar 3.2-5 Diagram Aplikasi

3.2.5 Proses Kerja Sistem Secara Keseluruhan (Flowchart Sistem)



Gambar 3.2-6 Flowchart Sistem Keseluruhan

3.2.6 Proses Teks Mining Metode LDA-AdaBoost (Flowchart Metode LDA-AdaBoost.MH)



Gambar 3.2-7 Flowchart Metode LDA-AdaBoost.MH

BAB IV IMPLEMENTASI SISTEM

Bab ini membahas implementasi dari perancangan sistem ERP 2015. Di dalamnya mencakup penjelasan lingkungan pengembangan sistem serta proses implementasi *distributed database*, *RBAC*, *multi-tenancy*, dan antarmuka pengguna.

4.1 Lingkungan Pengembangan Sistem

Lingkungan pengembangan sistem yang digunakan untuk mengembangkan Tugas Akhir ini dilakukan pada lingkungan dan kaskas sebagai berikut.

1. Database yang digunakan pada server adalah MySQL.
2. 1 PC untuk pengembangan User menggunakan Intel® Core™ i3-4010U @1.70GHz , RAM 4GB dengan Sistem Operasi Windows 7 Ultimate.
3. 1 PC untuk uji coba menggunakan Intel® Core™ i3-4010U @1.70GHz , RAM 4GB dengan Sistem Operasi Windows 7 Ultimate.
4. Server yang digunakan Xeon E5-2620 v2 12 Cores x 2.1 GHz dengan RAM 32GB DDR3.
5. Draw.io untuk pembuatan diagram dan sublime sebagai teks editor.
6. Python v2.7.12 untuk bahasa pemrograman pembuatan metode *LDA-AdaBoost.MH* dalam teks mining.
7. Mozilla Firefox 53.0.3 sebagai antarmuka untuk pengujian aplikasi pengguna.

4.2 Implementasi Program pada Pemeringkatan Universitas di Indonesia

Pada bagian ini akan dijelaskan mengenai implementasi yang terdapat pada Pemeringkatan Universitas di Indonesia untuk program pemeringkatan menggunakan metode *LDA-*

AdaBoost.MH. Secara rinci implementasi tersebut dijabarkan sebagai berikut:

4.2.1 Preprocessing

Sebelum data abstrak paper diproses oleh LDA-AdaBoost.MH, dilakukanlah preprocessing. Preprocessing ini bertujuan untuk mempermudah proses klasifikasi LDA-AdaBoost.MH dengan cara membuang puntuasi (*remove punctuation*), membuang kata yang tidak penting (*stop-words*), dan memecah kata (tokenisasi). Kode untuk melakukan preprocessing ditunjukkan oleh kode sumber A1.

4.2.2 Pembaruan Korpus

Pembaruan Korpus dilakukan untuk menyeimbangkan jumlah kata dalam korpus awal dengan jumlah kata dalam data uji coba abstrak jurnal akademik. Hal ini bertujuan untuk memberikan peningkatan akurasi dalam proses klasifikasi. Langkah kerja pembaruan korpus adalah dengan mencari kata yang cocok antara data uji coba abstrak jurnal akademik dengan korpus awal, kemudian ambil kata sebelum dan sesudah dari kata yang cocok tersebut untuk dijadikan tambahan kata dalam korpus yang baru. Kode tampilan yang dimaksud ditunjukkan oleh kode sumber A2.

4.2.3 Klasifikasi Term Frequency

Klasifikasi TF dilakukan untuk memberikan kata yang mewakili dokumen abstrak jurnal akademik berdasarkan frekuensi yang muncul. Hal ini dilakukan guna memberikan label pada masing-masing abstrak jurnal akademik masing-masing universitas yang akan dilakukan oleh LDA-AdaBoost.MH nanti.

Kode untuk menampilkan proses tersebut ditunjukkan pada kode sumber A3.

4.2.4 **Klasifikasi *LDA-AdaBoost.MH***

Klasifikasi *LDA-AdaBoost.MH* ini dilakukan untuk klasifikasi lebih lanjut dari pemrosesan TF sebelumnya. Hal ini ditujukan untuk memberikan label dan nilai probabilitas abstrak jurnal akademik masing-masing yang kemudian dijadikan penilaian pada reputasi akademik. Kode untuk menampilkan seluruh proses tersebut ditunjukkan pada kode sumber A4.

4.3 **Implementasi Basis Data**

Implementasi Basis Data ini dilakukan untuk menyimpan data dari semua proses pemeringkatan perguruan tinggi, yang kemudian akan dihubungkan/ditampilkan ke antarmuka pengguna. Kode untuk menampilkan seluruh proses implementasi basis data dapat ditunjukkan pada kode sumber A5.

4.4 **Implementasi Antarmuka Pengguna**

4.4.1 **Antarmuka Melihat Halaman Utama**

Pada antarmuka ini Pengguna dapat melihat halaman utama yang terdiri dari 6 bagian yaitu pencarian, sekilas informasi pemeringkatan, metode penilaian, peta (peringkat berdasar provinsi), lihat peringkat masing-masing metode (dengan *tab* custom untuk mengisi bobot sendiri), dan tentang kami. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.1, Gambar A.2, Gambar A.3, Gambar A.4, Gambar A.5, dan Gambar A.6.

4.4.2 Antarmuka Hasil Pencarian Universitas

Pada antarmuka ini Pengguna dapat melihat hasil pencarian universitas yang diinginkan. Pengguna dapat melihat detail informasi universitas tersebut dengan memilih tombol “lihat detail” dibagian kanan. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.7.

4.4.3 Antarmuka Detail Informasi Universitas

Pada antarmuka ini Pengguna dapat melihat detail informasi dari universitas yang telah dipilih. Informasi tersebut terdiri dari grafik pemeringkatan 9 metode universitas tersebut, alamat website universitas secara detail, jumlah mahasiswa, status akreditasi, jumlah dosen, dan informasi jurnal dosen. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.8.

4.4.4 Antarmuka Hasil Pemeringkatan Universitas

Pada antarmuka ini Pengguna dapat melihat hasil pemeringkatan universitas sesuai metode yang telah dipilih. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.9.

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Bab ini memberikan bahasan hasil pada sistem yang dikembangkan. Bab ini membahas data yang digunakan, penggunaan data terhadap sistem pada perangkat lunak dan uji coba fungsionalitas sistem pada perangkat lunak.

5.1 Lingkungan Uji Coba

Lingkungan uji coba adalah lingkungan yang digunakan untuk menguji implementasi pembuatan sistem pada tugas akhir ini. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

1. Perangkat keras
 - a. Prosesor: Intel® Core™ i3 CPU @ 1.70GHz
 - b. Memori (RAM): 4 GB
 - c. Tipe sistem: 32-bit sistem operasi
2. Perangkat lunak
 - a. Sistem operasi: Windows 7 Ultimate
 - b. Perangkat pengembang: Python-2.7.11

5.2 Data Uji Coba

Pengujian tugas akhir ini memiliki beberapa data uji coba yang dijelaskan pada subbab ini.

5.2.1 Data Korpus Tingkat Kesiapterapan Teknologi

Data Korpus Tingkat Kesiapterapan Teknologi merupakan kumpulan kata-kata yang diperoleh melalui pengembangan korpus *taxonomy bloom* secara manual dengan cara membaca satu-persatu kata-kata mana yang dapat mewakili tiap-tiap Tingkat Kesiapterapan Teknologi. Data uji ini bertujuan untuk acuan klasifikasi dari Data Abstrak Jurnal Akademik menggunakan metode teks mining *LDA-AdaBoost.MH*.

5.2.1.1 *Data Korpus Taxonomy Bloom*

Data Korpus Taxonomy Bloom merupakan framework kumpulan kata-kata yang dikelompokkan ke 6 kategori dengan beberapa sub-kategori dalam bidang edukasi. 6 kategori tersebut diantaranya:

- Knowledge

Berisi kemampuan dalam mengenali dan mengingat peristilahan, definisi, fakta-fakta, gagasan, pola, urutan, metodologi dan prinsip dasar. Contohnya, ketika dimintai penjelasan tentang manajemen kualitas, orang yang berada di level ini bisa menguraikan dengan baik definisi dari kualitas, karakteristik produk yang berkualitas, serta standar kualitas minimum untuk produk. Dalam kategori yang paling dasar ini, terdapat 73 kata yang dapat mewakili.

- Comprehension

Berisi kemampuan dalam mendemonstrasikan fakta dan gagasan mengelompokkan dengan mengorganisir, membandingkan, menerjemahkan, memaknai, memberi deskripsi, dan menyatakan gagasan utama. Orang yang berada dalam level ini sanggup membandingkan manfaat mengkonsumsi apel dan jeruk terhadap kesehatan. Dalam kategori ini, terdapat 49 kata yang dapat mewakili.

- Application

Di tingkat ini, seseorang memiliki kemampuan untuk menerapkan gagasan, prosedur, metode, rumus, dan teori dalam kondisi kerja. Sebagai contoh, ketika diberi informasi tentang penyebab meningkatnya reject pada produksi, seseorang yang berada di tingkat aplikasi ini akan mampu merangkum dan menggambarkan penyebab turunnya kualitas dalam bentuk diagram. Dalam kategori ini, terdapat 62 kata yang dapat mewakili.

- Analysis

Di tingkat analisis, seseorang akan mampu menganalisis informasi yang masuk dan menstrukturkan

informasi ke dalam bagian yang lebih kecil untuk mengenali pola hubungannya, dan mampu mengenali serta membedakan faktor penyebab dan akibat dari sebuah skenario yg rumit. Dalam tingkat ini, terdapat 62 kata yang dapat mewakili.

- Synthesis

Satu tingkat di atas analisis, seseorang di tingkat sintesis ini akan mampu menjelaskan struktur atau pola dari sebuah skenario yang sebelumnya tidak terlihat, dan mampu mengenali data atau informasi yang harus didapat untuk menghasilkan solusi yg dibutuhkan. Dalam tingkat ini, terdapat 73 kata yang dapat mewakili.

- Evaluation

Berisi kemampuan untuk memberikan penilaian terhadap solusi, gagasan, dan metodologi dengan menggunakan kriteria yang cocok atau standar yang ada untuk memastikan nilai efektivitas atau manfaatnya. Dalam tingkat ini, terdapat 59 kata yang dapat mewakili.

5.2.1.2 *Data Korpus dengan POS-Tagger*

Data Korpus TKT yang sudah dikembangkan dari *Taxonomy Bloom*, dikembangkan lagi dengan metode *POS-Tagger*. Hal ini ditujukan untuk meningkatkan akurasi dengan cara memberikan perbandingan yang sepadan antara jumlah kata dalam korpus dengan jumlah kata dalam data uji coba abstrak jurnal akademik. Cara kerja dari *POS-Tagger* ini adalah mencocokkan kata dalam data uji coba abstrak jurnal dengan korpus TKT *Taxonomy Bloom* saat ini. Jika terdapat satu kata yang cocok, maka diambil kata sebelum dan sesudah dari kata tersebut yang kemudian ditambahkan ke dalam korpus yang baru. Dengan pengembangan korpus *POS-Tagger* ini, didapatkan total kata 2063 untuk korpus yang baru.

5.2.2 Data Abstrak Jurnal Akademik

Data Abstrak Jurnal Akademik merupakan kumpulan abstrak dari *paper* akademisi masing-masing universitas yang diurutkan dari sitasi tertinggi sampai terendah. Data uji coba ini digunakan untuk memenuhi salah satu indikator reputasi akademik pengganti survey manual yang dilakukan oleh QS dengan cara melakukan teks mining metode *LDA-AdaBoost.MH*.

Data uji coba ini berisi 50 abstrak jurnal akademik 16 universitas peringkat tertinggi mengacu pada QS dengan total kata 139919.

5.2.3 Data *Science and Technology Index* (SINTA)

SINTA merupakan website resmi dari Kementerian Ristekdikti yang menyediakan informasi detail mengenai jurnal akademik universitas di Indonesia. Data uji coba yang diambil dari SINTA adalah data mengenai jumlah jurnal akademik masing-masing universitas beserta sitasinya dalam Scopus dan nilai *h-index*/*i10-index*. Hal ini dikarenakan terbatasnya akses ke dalam website Scopus yang mengharuskan pembuatan *member* secara berbayar.

Beberapa konten yang dapat diambil dari website SINTA ini antara lain:

1. Authors, merupakan jumlah penulis penelitian yang terindex oleh Google Scholar dan Scopus.
2. Google Citation, merupakan jumlah sitasi dari semua author dalam perguruan tinggi yang terindex oleh Google Scholar.
3. Google Documents, merupakan jumlah dokumen penelitian yang terindex Google Scholar.
4. Google Scholar Indexed, merupakan jumlah *h-index* dan *i10-index* akademisi yang terindex Google Scholar.
5. Scopus Indexed, merupakan jumlah *h-index* dan *i10-index* akademisi yang terindex Scopus.

6. Journal Articles, merupakan jumlah paper penelitian dari akademisi yang terindex Scopus.
7. Articles in Scopus, merupakan artikel penelitian dari akademisi yang terindex Scopus.
8. Non Articles in Scopus, merupakan non artikel yang terindex Scopus.

Sebagai catatan bahwa data yang disediakan oleh website SINTA untuk Tugas Akhir ini kurang valid dikarenakan basis data SINTA tidak diperbarui secara berkala. Data SINTA yang digunakan pada Tugas Akhir ini adalah data periode 2016-2017. Data ini perlu dilakukan validasi terhadap periode terbaru setiap tahunnya. Alasan menggunakan data dari SINTA ini adalah karena data yang disediakan mudah didapat dan sesuai dengan variabel indikator penilaian Ground Truth QS World University Rankings yaitu jumlah sitasi. Jika data yang digunakan pada SINTA diperbarui maka otomatis hasil pemeringkatan Perguruan Tinggi pada Tugas Akhir ini akan berubah juga sesuai kondisi data *real time* dengan bobot indikator sebesar 20%.

5.2.4 Data Jumlah Mahasiswa dan Dosen

Data uji coba ini diperoleh melalui website resmi PDDIKTI. Data yang diambil adalah jumlah sekaligus rasio mahasiswa dan dosen masing-masing universitas yang terdaftar dalam PDDIKTI.

5.2.5 Pengujian Fungsionalitas

5.2.5.1 Pengujian Fitur Mengelola Item Master

Tabel 5.1 Tabel Pengujian Fitur Melihat Peringkat Universitas

ID	UJ-UC-001
Referensi Kasus Penggunaan	UC-001
Nama	Pengujian Fitur Melihat Peringkat Universitas
Tujuan Pengujian	Menguji kemampuan aplikasi menampilkan peringkat universitas

Skenario 1	Pengguna melihat peringkat universitas
Kondisi Awal	Aktor berada pada halaman utama situs pemeringkatan universitas bagian lihat ranking
Data Uji	Inputan data indikator peringkat universitas
Langkah Pengujian	Aktor memilih salah satu metode untuk pemeringkatan universitas.
Hasil Yang Diharapkan	Menampilkan peringkat universitas sesuai dengan metode yang dipilih
Hasil Yang Didapat	Peringkat universitas berhasil ditampilkan sesuai metode yang dipilih
Hasil Pengujian	Berhasil
Kondisi Akhir	Aktor berada pada halaman peringkat universitas sesuai metode yang dipilih

➤ Skenario 1

Lampiran Gambar A.10 menampilkan halaman utama bagian lihat ranking. Skenario ini dimulai ketika aktor memilih salah satu dari 9 metode yang telah disediakan oleh sistem. Setelah aktor memilih salah satu metode tersebut, sistem akan memulai perhitungan sesuai dengan indikatornya masing-masing untuk mendapatkan hasil pemeringkatan universitas di Indonesia. Hasil akhir dari skenario ini adalah sistem menampilkan halaman berupa peringkat universitas yang dapat dilihat pada Lampiran Gambar A.11.

5.2.5.2 Pengujian Fitur Memberikan Bobot Kriteria

Tabel 5.2 Tabel Pengujian Fitur Memberikan Bobot Kriteria

ID	UJ-UC-002
Referensi Kasus Penggunaan	UC-002
Nama	Pengujian Fitur Memberikan Bobot Kriteria
Tujuan Pengujian	Menguji kemampuan aplikasi dalam menerima dan mengelola inputan bobot sesuai keinginan pengguna untuk mendapatkan peringkat universitas yang diinginkan
Skenario 1	Pengguna memberikan bobot kriteria
Kondisi Awal	Aktor berada pada halaman utama bagian lihat ranking dengan <i>tab custom</i>

Data Uji	Inputan data bobot kriteria indikator
Langkah Pengujian	Aktor memasukkan bobot kriteria ke indikator-indikator yang disediakan
Hasil Yang Diharapkan	Menampilkan peringkat universitas sesuai dengan bobot kriteria yang diberikan oleh pengguna
Hasil Yang Didapat	Peringkat universitas berhasil ditampilkan sesuai dengan bobot kriteria yang diberikan pengguna
Hasil Pengujian	Berhasil
Kondisi Akhir	Aktor berada pada halaman peringkat universitas sesuai bobot kriteria yang diberikan

➤ Skenario 1

Lampiran Gambar A.12 menampilkan halaman utama bagian lihat ranking dengan *tab custom*. Skenario ini dimulai ketika aktor memberikan bobot kriteria ke indikator-indikator yang diinginkan. Kemudian sistem akan mengolah bobot tersebut sehingga dapat menampilkan peringkat universitas sesuai bobot yang diinginkan pengguna. Hasil akhir dari skenario ini adalah sistem menampilkan halaman peringkat universitas sesuai dengan bobot dari pengguna yang dapat dilihat pada Lampiran Gambar A.13.

5.2.5.3 Pengujian Fitur Mencari Universitas

Tabel 5.3 Tabel Pengujian Fitur Mencari Universitas

ID	UJ-UC-003
Referensi Kasus Penggunaan	UC-003
Nama	Pengujian Fitur Mencari Universitas
Tujuan Pengujian	Menguji kemampuan aplikasi menampilkan universitas yang dicari pengguna
Skenario 1	<i>Pengguna memasukkan nama universitas ke kotak pencarian</i>
Kondisi Awal	Aktor berada pada halaman utama bagian pencarian
Data Uji	Inputan data nama universitas
Langkah Pengujian	Pengguna memasukkan nama universitas yang akan dicari ke kotak pencarian yang telah disediakan oleh sistem pada halaman utama bagian pencarian.

Hasil Yang Diharapkan	Menampilkan hasil pencarian sesuai dengan nama universitas yang dimasukkan oleh pengguna dalam kotak pencarian.
Hasil Yang Didapat	Hasil universitas berhasil ditampilkan sesuai dengan nama universitas yang dicari pengguna.
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman hasil pencarian universitas sesuai nama yang dimasukkan pengguna dalam kotak pencarian.

➤ Skenario 1

Lampiran Gambar A.14 menampilkan halaman utama bagian pencarian. Skenario ini dimulai ketika aktor memasukkan nama universitas yang ingin dicari dan menekan tombol *enter* pada keyboard. Kemudian sistem akan memulai mencocokkan nama universitas yang dimasukkan pengguna dengan data yang ada. Ketika nama universitas yang dicari telah ditemukan dalam data sistem, sistem akan menampilkan data-data yang cocok dengan inputan nama universitas dari pengguna. Hasil akhir dari skenario ini adalah sistem menampilkan hasil pencarian yang cocok dengan inputan nama universitas dari pengguna yang dapat dilihat pada Lampiran Gambar A.15.

5.2.5.4 Pengujian Fitur Mencari Informasi Peringkat Universitas Sesuai Provinsi

Tabel 5.4 Tabel Pengujian Fitur Mencari Informasi Peringkat Universitas Sesuai Provinsi

ID	UJ-UC-004
Referensi Kasus Penggunaan	UC-004
Nama	Pengujian Fitur Mencari Informasi Peringkat Universitas Sesuai Provinsi
Tujuan Pengujian	Menguji kemampuan aplikasi menampilkan peringkat universitas sesuai provinsi yang dipilih pengguna
Skenario 1	<i>Aktor memilih provinsi pada peta yang disediakan sistem</i>
Kondisi Awal	Aktor berada pada halaman utama bagian peta
Data Uji	Inputan data peringkat universitas sesuai provinsi

Langkah Pengujian	Pengguna memilih provinsi yang diinginkan pada peta yang telah disediakan oleh sistem pada halaman utama bagian peta.
Hasil Yang Diharapkan	Menampilkan peringkat universitas sesuai provinsi yang dipilih oleh pengguna.
Hasil Yang Didapat	Peringkat universitas berdasarkan provinsi berhasil ditampilkan sesuai dengan yang dipilih oleh pengguna
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja disunting.
<i>Skenario 2</i>	<i>Aktor menghapus data Good Issue</i>
Kondisi Awal	Aktor berada pada halaman peringkat universitas berdasarkan provinsi yang telah dipilih.

➤ Skenario 1

Lampiran Gambar A.16 menampilkan halaman utama bagian peta. Skenario ini dimulai ketika aktor memilih provinsi yang diinginkan untuk pemeringkatan universitas pada peta yang telah disediakan oleh sistem. Hasil akhir dari skenario ini adalah sistem menampilkan hasil pemeringkatan universitas sesuai provinsi yang dipilih oleh pengguna yang dapat dilihat pada Lampiran Gambar A.17.

5.2.5.5 Pengujian Fitur Melihat Detail Informasi Universitas

Tabel 5.5 Tabel Pengujian Fitur Melihat Detail Informasi Universitas

ID	UJ-UC-005
Referensi Kasus Penggunaan	UC-005
Nama	Pengujian Fitur Melihat Detail Informasi Universitas
Tujuan Pengujian	Menguji kemampuan aplikasi menampilkan detail informasi universitas
<i>Skenario 1</i>	<i>Pengguna melihat detail informasi universitas</i>
Kondisi Awal	Aktor berada pada halaman hasil pencarian universitas yang diinginkan
Data Uji	Inputan data detail informasi universitas
Langkah Pengujian	Aktor menekan tombol lihat detail pada sebelah kanan dari universitas yang diinginkan.

Hasil Yang Diharapkan	Sistem menampilkan detail informasi dari universitas yang telah dipilih oleh pengguna.
Hasil Yang Didapat	Detail informasi universitas berhasil ditampilkan sesuai yang diinginkan pengguna.
Hasil Pengujian	Berhasil
Kondisi Akhir	Aktor berada pada halaman detail informasi universitas yang dipilih.
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman utama Transfer Posting.

➤ Skenario 1

Lampiran Gambar A.18 menampilkan halaman hasil pencarian nama universitas. Skenario ini dimulai ketika aktor menekan tombol lihat detail pada bagian kanan dari universitas yang dipilih. Setelah itu sistem akan menampilkan halaman detail informasi sesuai keinginan dari pengguna yang dapat dilihat pada Lampiran Gambar A.19.

5.2.5.6 Pengujian Fitur Mengelola Data Informasi Universitas

Tabel 5.6 Tabel Pengujian Fitur Mengelola Data Informasi Universitas

ID	UJ-UC-006
Referensi Kasus Penggunaan	UC-006
Nama	Pengujian Fitur Mengelola Data Informasi Universitas
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data informasi universitas
Skenario 1	<i>Aktor menambahkan data informasi universitas</i>
Kondisi Awal	Aktor berada pada halaman kontrol data informasi universitas
Data Uji	Inputan data informasi universitas
Langkah Pengujian	Administrator menambahkan salah satu data informasi universitas yang ada pada sistem dan menyimpannya kembali ke basis data.
Hasil Yang Diharapkan	Data berhasil ditambahkan dan disimpan kembali ke basis data.
Hasil Yang Didapat	Data yang ditambahkan tersimpan di basis data

Hasil Pengujian	Berhasil.
Kondisi Akhir	Administrator kembali berada pada halaman kontrol data informasi universitas.
Skenario 2	Aktor menyunting data informasi universitas
Kondisi Awal	Aktor berada pada halaman utama kontrol data informasi universitas.
Data Uji	Data informasi universitas yang disunting
Langkah Pengujian	Administrator menyunting data yang ada pada sistem, lalu melakukan konfirmasi perubahan data tersebut dan data akan diperbarui oleh sistem.
Hasil Yang Diharapkan	Data yang disunting berhasil dirubah dan tersimpan di basis data.
Hasil Yang Didapat	Data yang disunting tersimpan di basis data.
Hasil Pengujian	Berhasil.
Kondisi Akhir	Administrator kembali berada pada halaman kontrol data informasi universitas.
Skenario 3	Aktor menghapus data informasi universitas
Kondisi Awal	Aktor berada pada halaman kontrol data informasi universitas
Data Uji	Data informasi universitas yang dihapus
Langkah Pengujian	Administrator menghapus data yang ada pada sistem, lalu melakukan konfirmasi penghapusan data tersebut dan data akan diperbarui oleh sistem.
Hasil Yang Diharapkan	Data yang dihapus akan terhapus di basis data.
Hasil Yang Didapat	Data berhasil dihapus.
Hasil Pengujian	Berhasil.
Kondisi Akhir	Administrator kembali berada pada halaman kontrol data informasi universitas.

➤ Skenario 1

Lampiran Gambar A.20 menampilkan halaman kontrol data informasi universitas. Skenario ini dimulai ketika aktor menambahkan data baru informasi universitas yang dapat dilihat pada Lampiran Gambar A.21. Setelah itu sistem akan menampilkan halaman kontrol data informasi universitas beserta penambahan data yang dilakukan oleh aktor.

➤ Skenario 2

Lampiran Gambar A.22 menampilkan halaman kontrol data informasi universitas. Skenario ini dimulai ketika aktor menyunting data informasi universitas yang dapat dilihat pada Lampiran Gambar A.23. Setelah itu sistem akan menampilkan halaman kontrol data informasi universitas beserta perubahan data yang dilakukan oleh aktor.

➤ Skenario 3

Lampiran Gambar A.24 menampilkan halaman kontrol data informasi universitas. Skenario ini dimulai ketika aktor menghapus data informasi universitas yang dapat dilihat pada Lampiran Gambar A.25. Setelah itu sistem akan menampilkan halaman kontrol data informasi universitas beserta penghapusan data yang dilakukan oleh aktor.

5.2.5.7 Pengujian Fitur Melihat Metode Penilaian

Tabel 5.7 Tabel Pengujian Fitur Melihat Metode Penilaian

ID	UJ-UC-007
Referensi Kasus Penggunaan	UC-007
Nama	Pengujian Fitur Melihat Metode Penilaian
Tujuan Pengujian	Menguji kemampuan aplikasi melihat metode penilaian
Skenario 1	<i>Pengguna melihat metode penilaian</i>
Kondisi Awal	Aktor berada pada halaman utama bagian metode penilaian
Data Uji	Inputan data indikator metode penilaian
Langkah Pengujian	Aktor menekan tombol lihat metode penilaian yang kemudian diarahkan ke halaman informasi metode penilaian
Hasil Yang Diharapkan	Menampilkan informasi metode penilaian
Hasil Yang Didapat	Informasi penilaian berhasil ditampilkan
Hasil Pengujian	Berhasil
Kondisi Akhir	Aktor berada pada halaman informasi metode penilaian

➤ Skenario 1

Lampiran Gambar A.26 menampilkan halaman utama bagian metode penilaian. Skenario ini dimulai ketika aktor menekan tombol lihat metode penilaian dan sistem menampilkan detail informasi metode penilaian yang dapat dilihat pada Lampiran Gambar A.27.

5.3 Skenario Pengujian

5.3.1 Preprocessing

Data abstrak jurnal akademik yang akan dilakukan preprocessing adalah berupa kumpulan 50 abstrak dengan sitasi terbanyak masing-masing perguruan tinggi. Data mentah tersebut diolah dengan membuang pungtuasinya (*remove punctuation*), membuang kata yang tidak penting (*stop-words*), dan memecah kata (tokenisasi). Hasil preprocessing untuk data-data mentah tersebut dapat dilihat pada Tabel 5.8.

Tabel 5.8 Hasil preprocessing

No.	Nama Perguruan Tinggi	Jumlah Kata Asli	Jumlah Kata Hasil Preprocessing
1.	Institut Pertanian Bogor	10539	6293
2.	Institut Teknologi Bandung	9287	5675
3.	Institut Teknologi Sepuluh Nopember	9630	6015
4.	Universitas 17 Agustus 1945	9146	5580
5.	Universitas Airlangga	9457	6009
6.	Universitas Bina Nusantara	7236	4432
7.	Universitas Brawijaya	8651	5469

8.	Universitas Diponegoro	8397	5327
9.	Universitas Gadjah Mada	10214	6279
10.	Universitas Indonesia	8823	5645
11.	Universitas Islam Indonesia	7897	4989
12.	Universitas Mataram	7527	4561
13.	Universitas Muhammadiyah Surakarta	10330	6117
14.	Universitas Padjadjaran	7774	5008
15.	Universitas Pelita Harapan	7204	4542
16.	Universitas Sebelas Maret	7807	5244

5.3.2 Pembaruan Korpus

Korpus awal yang digunakan merupakan hasil pembagian dari 6 tingkatan *Bloom's Taxonomy* menjadi 9 tingkatan sesuai dengan Tingkat Kesiapterapan Teknologi. Hasil akhir dari pembagian tersebut dapat dilihat pada Tabel 5.9.

Tabel 5.9 Hasil Korpus Awal

No.	Nama TKT	Jumlah Kata
1.	TKT 1	31
2.	TKT 2	24
3.	TKT 3	32
4.	TKT 4	15
5.	TKT 5	31
6.	TKT 6	24
7.	TKT 7	26
8.	TKT 8	30
9.	TKT 9	34

Pada tabel diatas dapat disimpulkan bahwa terdapat 31 kata yang dapat mewakili TKT 1, 24 kata mewakili TKT 2, 32 kata mewakili TKT 3, 15 kata mewakili TKT 4, 31 kata mewakili TKT

5, 24 kata mewakili TKT 6, 26 kata mewakili TKT 7, 30 kata mewakili TKT 8, dan 34 kata mewakili TKT 9.

Kemudian setelah mendapatkan korpus awal di atas, diperlukannya pembaruan korpus guna memperkaya perwakilan masing-masing kata pada TKT. Pembaruan korpus ini menggunakan metode POS-Tagging, yang cara kerjanya adalah dengan mengambil kata yang berdekatan(sebelum dan sesudah) sesuai dengan korpus TKT pada dataset abstrak jurnal akademik. Hasil pembaruan korpus TKT dapat dilihat pada Tabel 5.10.

Tabel 5.10 Hasil Pembaruan Korpus

No.	Nama TKT	Jumlah Kata Awal	Jumlah Kata Pembaruan
1.	TKT 1	31	520
2.	TKT 2	24	358
3.	TKT 3	32	507
4.	TKT 4	15	462
5.	TKT 5	31	321
6.	TKT 6	24	421
7.	TKT 7	26	236
8.	TKT 8	30	158
9.	TKT 9	34	258

5.3.3 Klasifikasi Term Frequency

Hasil dari preprosesing akan dilakukan 2 percobaan klasifikasi, menggunakan *term frequency* antara dataset abstrak jurnal akademik dengan korpus TKT awal dan korpus TKT pembaruan (postagging) guna mendapatkan label masing-masing perguruan tinggi.

Percobaan 1 adalah melakukan *term frequency* dataset abstrak jurnal akademik terhadap TKT awal (tanpa postagging). Hasil pelabelan ini dapat dilihat pada Tabel 5.11.

Tabel 5.11 Hasil Pelabelan TKT awal

No.	Nama	3 Label Teratas	Hasil Label Final
1.	IPB	T7, T8, T9	T9
2.	ITB	T7, T8, T9	T9

3.	ITS	T5, T8, T9	T9
4.	UNTAG	T7, T5, T8	T8
5.	UNAIR	T7, T8, T9	T7
6.	BINUS	T7, T5, T9	T9
7.	UB	T6, T8, T9	T9
8.	UNDIP	T6, T7, T9	T7
9.	UGM	T7, T8, T9	T9
10.	UI	T7, T8, T9	T9
11.	UIN	T6, T7, T9	T9
12.	UNRAM	T7, T5, T9	T7
13.	UMS	T7, T8, T9	T7
14.	UNPAD	T5, T8, T9	T5
15.	UPH	T7, T5, T9	T9
16.	UNS	T7, T5, T8	T8

Kemudian dilanjut dengan percobaan 2, yaitu proses *term frequency* terhadap TKT yang telah terbarukan(postagging). Hasil pelabelan ini dapat dilihat pada Tabel 5.12.

Tabel 5.12 Hasil Pelabelan TKT baru

No.	Nama	3 Label Teratas	Hasil Label Final
1.	IPB	T6, T7, T5	T6
2.	ITB	T6, T7, T5	T6
3.	ITS	T6, T7, T5	T5
4.	UNTAG	T6, T7, T5	T6
5.	UNAIR	T6, T7, T5	T7
6.	BINUS	T6, T7, T5	T6
7.	UB	T6, T7, T5	T5
8.	UNDIP	T6, T7, T5	T6
9.	UGM	T6, T7, T5	T7
10.	UI	T6, T7, T5	T7
11.	UIN	T6, T7, T5	T5
12.	UNRAM	T6, T7, T5	T5
13.	UMS	T6, T7, T5	T6
14.	UNPAD	T6, T7, T5	T7
15.	UPH	T6, T7, T5	T5
16.	UNS	T6, T7, T5	T5

5.3.4 Pembobotan Final Pemeringkatan

Pada tahap terakhir ini, akan didapat hasil probabilitas 3 label tertinggi masing-masing perguruan tinggi yang kemudian digunakan untuk pembobotan final reputasi akademik.

Untuk percobaan 1 menggunakan LDA terhadap korpus TKT awal(tanpa postagging) dalam perhitungan bobot masing-masing labelnya. Hasil percobaan ini dapat dilihat pada Tabel 5.13.

Tabel 5.13 Hasil Probabilitas LDA terhadap TKT awal

No.	Nama	Level TKT								
		1	2	3	4	5	6	7	8	9
1.	IPB	0	0	0	0	0	0	0,33	0,33	0,34
2.	ITB	0	0	0	0	0	0	0,31	0,34	0,35
3.	ITS	0	0	0	0	0	0	0,34	0,32	0,34
4.	UNTAG	0	0	0	0	0	0	0,35	0,32	0,33
5.	UNAIR	0	0	0	0	0	0	0,35	0,32	0,33
6.	BINUS	0	0	0	0	0,33	0	0,30	0	0,37
7.	UB	0	0	0	0	0	0,32	0	0,33	0,34
8.	UNDIP	0	0	0	0	0	0,30	0,36	0	0,34
9.	UGM	0	0	0	0	0	0	0,33	0,33	0,34
10.	UI	0	0	0	0	0	0	0,33	0,33	0,34
11.	UII	0	0	0	0	0	0,33	0,32	0	0,35
12.	UNRAM	0	0	0	0	0,31	0	0,37	0	0,32
13.	UMS	0	0	0	0	0	0	0,36	0,30	0,33
14.	UNPAD	0	0	0	0	0,35	0	0	0,32	0,33
15.	UPH	0	0	0	0	0,33	0	0,31	0	0,36
16.	UNS	0	0	0	0	0,34	0	0,30	0	0,36

Sesuai dengan hasil probabilitas LDA di atas, akan dilakukan pembobotan untuk peringkat perguruan tingginya. Hasil peringkat, beda peringkat terhadap peringkat QS dan toleransi ke-16 perguruan tinggi di atas dapat dilihat pada Tabel 5.14.

Tabel 5.14 Hasil peringkat perguruan tinggi LDA terhadap TKT awal

Nama	Peringkat Ground Truth	Peringkat Sistem tanpa postagging)	Beda	Toleransi (<=3)
BINUS	13	12	1	1

IPB	6	3	3	1
ITB	1	1	0	1
ITS	9	10	1	1
UB	8	7	1	1
UGM	3	2	1	1
UI	2	4	2	1
UIN	11	9	2	1
UMS	4	6	2	1
UNAIR	5	5	0	1
UNDIP	7	8	1	1
UNPAD	10	11	1	1
UNRAM	16	14	2	1
UNS	15	16	1	1
UNTAG	12	15	3	1
UPH	14	13	1	1

Kemudian untuk percobaan 2, menggunakan LDA terhadap korpus TKT baru(postagging) dalam perhitungan bobot masing-masing labelnya. Hasil percobaan ini dapat dilihat pada Tabel 5.15.

Tabel 5.15 Hasil Pembobotan LDA terhadap TKT baru

No.	Nama	Level TKT								
		1	2	3	4	5	6	7	8	9
1.	IPB	0	0	0	0	0,32	0,35	0,33	0	0
2.	ITB	0	0	0	0	0,32	0,34	0,34	0	0
3.	ITS	0	0	0	0	0,35	0,32	0,33	0	0
4.	UNTAG	0	0	0	0	0,31	0,41	0,28	0	0
5.	UNAIR	0	0	0	0	0,30	0,35	0,35	0	0
6.	BINUS	0	0	0	0	0,33	0,34	0,34	0	0
7.	UB	0	0	0	0	0,35	0,33	0,32	0	0
8.	UNDIP	0	0	0	0	0,34	0,35	0,31	0	0
9.	UGM	0	0	0	0	0,32	0,31	0,37	0	0
10.	UI	0	0	0	0	0,34	0,30	0,36	0	0
11.	UIN	0	0	0	0	0,36	0,31	0,33	0	0
12.	UNRAM	0	0	0	0	0,38	0,28	0,34	0	0
13.	UMS	0	0	0	0	0,33	0,35	0,32	0	0
14.	UNPAD	0	0	0	0	0,33	0,34	0,33	0	0
15.	UPH	0	0	0	0	0,34	0,32	0,34	0	0
16.	UNS	0	0	0	0	0,37	0,30	0,33	0	0

Sesuai dengan hasil probabilitas LDA terhadap TKT baru di atas, akan dilakukan pembobotan untuk peringkat perguruan tingginya. Hasil peringkat, beda peringkat terhadap peringkat QS dan toleransi ke-16 perguruan tinggi di atas dapat dilihat pada Tabel 5.16.

Tabel 5.16 Hasil Peringkat Perguruan Tinggi LDA terhadap TKT baru

Nama	Peringkat Ground Truth	Peringkat Sistem (pos tagging 1)	Beda	Toleransi (≤ 3)
BINUS	13	10	3	1
IPB	6	5	1	1
ITB	1	4	3	1
ITS	9	11	2	1
UB	8	13	5	0
UGM	3	2	1	1
UI	2	3	1	1
UIN	11	14	3	1
UMS	4	6	2	1
UNAIR	5	1	4	0
UNDIP	7	7	0	1
UNPAD	10	9	1	1
UNRAM	16	16	0	1
UNS	15	15	0	1
UNTAG	12	12	0	1
UPH	14	8	6	0

Untuk percobaan 3 menggunakan LDA-AdaBoost.MH terhadap korpus TKT awal(tanpa postagging) dalam perhitungan bobot masing-masing labelnya. Hasil percobaan ini dapat dilihat pada Tabel 5.17.

Tabel 5.17 Hasil Probabilitas LDA-AdaBoost.MH terhadap TKT awal

No.	Nama	Level TKT								
		1	2	3	4	5	6	7	8	9
1.	IPB	0	0	0	0	0	0	0.30	0.30	0.40
2.	ITB	0	0	0	0	0	0	0.32	0.30	0.38
3.	ITS	0	0	0	0	0.31	0	0	0.30	0.39

4.	UNTAG	0	0	0	0	0.34	0	0.33	0	0.33
5.	UNAIR	0	0	0	0	0	0	0.30	0.29	0.41
6.	BINUS	0	0	0	0	0.32	0	0.29	0	0.38
7.	UB	0	0	0	0	0	0.27	0	0.32	0.41
8.	UNDIP	0	0	0	0	0	0.26	0.31	0	0.43
9.	UGM	0	0	0	0	0	0	0.30	0.32	0.38
10.	UI	0	0	0	0	0	0	0.30	0.31	0.39
11.	UIN	0	0	0	0	0	0.33	0.27	0.31	0.41
12.	UNRAM	0	0	0	0	0.31	0	0.28	0.31	0.41
13.	UMS	0	0	0	0	0	0	0.30	0.32	0.39
14.	UNPAD	0	0	0	0	0.30	0	0	0.30	0.40
15.	UPH	0	0	0	0	0.30	0	0.31	0	0.39
16.	UNS	0	0	0	0	0.34	0	0.33	0	0.33

Sesuai dengan hasil probabilitas LDA-AdaBoost.MH di atas, akan dilakukan pembobotan untuk peringkat perguruan tingginya. Hasil peringkat, beda peringkat terhadap peringkat QS dan toleransi ke-16 perguruan tinggi di atas dapat dilihat pada Tabel 5.18.

Tabel 5.18 Hasil peringkat perguruan tinggi LDA-AdaBoost.MH terhadap TKT awal

Nama	Peringkat <i>Ground Truth</i> QS	Peringkat Sistem tanpa postagging)	Beda	Toleransi (≤ 3)
ITB	1	6	5	0
UI	2	4	2	1
UGM	3	5	2	1
UMS	4	3	1	1
UNAIR	5	1	4	0
IPB	6	2	4	0
UNDIP	7	8	1	1
UB	8	7	1	1
ITS	9	11	2	1
UNPAD	10	10	0	1
UIN	11	9	2	1
UNTAG	12	16	4	0
BINUS	13	13	0	1
UPH	14	14	0	1
UNS	15	15	0	1

UNRAM	16	12	4	0
-------	----	----	---	---

Kemudian untuk percobaan 4, menggunakan LDA-AdaBoost.MH terhadap korpus TKT baru(postagging) dalam perhitungan bobot masing-masing labelnya. Hasil percobaan ini dapat dilihat pada Tabel 5.19.

Tabel 5.19 Hasil Pembobotan LDA-AdaBoost.MH terhadap TKT baru

No.	Nama	Level TKT								
		1	2	3	4	5	6	7	8	9
1.	IPB	0	0	0	0	0,32	0,36	0,29	0	0
2.	ITB	0	0	0	0	0,35	0,36	0,29	0	0
3.	ITS	0	0	0	0	0,36	0,35	0,29	0	0
4.	UNTAG	0	0	0	0	0,34	0,39	0,27	0	0
5.	UNAIR	0	0	0	0	0,34	0,36	0,30	0	0
6.	BINUS	0	0	0	0	0,35	0,36	0,29	0	0
7.	UB	0	0	0	0	0,36	0,35	0,28	0	0
8.	UNDIP	0	0	0	0	0,36	0,36	0,28	0	0
9.	UGM	0	0	0	0	0,35	0,35	0,31	0	0
10.	UI	0	0	0	0	0,36	0,34	0,31	0	0
11.	UII	0	0	0	0	0,37	0,34	0,29	0	0
12.	UNRAM	0	0	0	0	0,38	0,33	0,29	0	0
13.	UMS	0	0	0	0	0,35	0,36	0,29	0	0
14.	UNPAD	0	0	0	0	0,35	0,36	0,29	0	0
15.	UPH	0	0	0	0	0,36	0,35	0,29	0	0
16.	UNS	0	0	0	0	0,37	0,34	0,29	0	0

Sesuai dengan hasil probabilitas LDA terhadap TKT baru di atas, akan dilakukan pembobotan untuk peringkat perguruan tingginya. Hasil peringkat, beda peringkat terhadap peringkat QS dan toleransi ke-16 perguruan tinggi di atas dapat dilihat pada Tabel 5.20.

Tabel 5.20 Hasil Peringkat Perguruan Tinggi LDA-AdaBoost.MH terhadap TKT baru

Nama	Peringkat Ground Truth	Peringkat Sistem (pos tagging 1)	Beda	Toleransi (≤ 3)
ITB	1	4	3	1

UI	2	3	1	1
UGM	3	2	1	1
UMS	4	7	3	1
UNAIR	5	1	4	0
IPB	6	6	0	1
UNDIP	7	12	5	0
UB	8	13	5	0
ITS	9	10	1	1
UNPAD	10	9	1	1
UJI	11	14	3	1
UNTAG	12	11	1	1
BINUS	13	5	8	0
UPH	14	8	6	0
UNS	15	15	0	1
UNRAM	16	16	0	1

Untuk ringkasan keseluruhan, dapat dilihat pada Tabel 5.21.

Tabel 5.21 Ringkasan Performa LDA dan LDA-AdaBoost.MH

Lingkup	Indikator	Percobaan 1(LDA+TK T awal)	Percobaan 2(LDA+TK T baru)	Percobaan 3(LDA-AdaBoost+TK T awal)
Indikator Reputasi Akademik	Beda (error)	22	32	32
	Toleransi	16/16	13/16	11/16
	Prosentase toleransi	100%	81,25%	68,75%
	Keterangan prosentase toleransi	100% peringkat masuk dalam toleransi	81,25% peringkat masuk dalam toleransi	68,75% peringkat masuk dalam toleransi
	Similarity Pearson Correlation	93,8%	83,9%	84,1%
Sistem secara	Beda (error)	16	18	18

keseluruhan (all indicator)	Toleransi	10/11	10/11	10/11
	Prosentase toleransi	90,91%	90,91%	90,91%
	Keterangan prosentase toleransi	90,91% peringkat masuk dalam toleransi	90,91% peringkat masuk dalam toleransi	90,91% peringkat masuk dalam toleransi
	Similarity Pearson Correlation	79,091%	77,273%	86,4%

5.4 Evaluasi Pengujian

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada Tabel 5.21. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan baik. Sehingga bisa didapat kesimpulan bahwa fungsionalitas dari program telah bekerja sesuai dengan yang diharapkan.

Tabel 5.22 Rangkuman Hasil Pengujian

ID	Nama	Skenario	Hasil
UJ-UC-001	Melihat Peringkat Universitas	Skenario 1	Berhasil
UJ-UC-002	Memberikan Bobot Kriteria	Skenario 1	Berhasil
UJ-UC-003	Mencari Universitas	Skenario 1	Berhasil
UJ-UC-004	Mencari Informasi Peringkat Universitas Sesuai Provinsi	Skenario 1	Berhasil
UJ-UC-005	Melihat Detail Informasi Universitas	Skenario 1	Berhasil
UJ-UC-006	Mengelola Data Informasi Universitas	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
UJ-UC-007	Melihat Metode Penilaian	Skenario 1	Berhasil

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang dapat diambil dari pengerjaan serta saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1 Kesimpulan

Dari hasil pengamatan selama pengerjaan Tugas Akhir, dapat disimpulkan sebagai berikut:

1. Data untuk pengolahan peringkat universitas dapat didapatkan secara online melalui metode teks mining.
2. Indikator reputasi akademik dalam pemeringkatan universitas pada QS yang didapatkan melalui survey manual, terbukti dapat digantikan oleh metode teks mining *LDA-AdaBoost.MH* yang dilakukan secara otomatis dengan tingkat akurasi 86,4%.
3. Kinerja sistem secara keseluruhan dapat menyamai ground truth QS World University Ranking 2016-2017.
4. Metode teks mining dapat dikembangkan lagi menggunakan teknik POS Tagging untuk memperkaya corpus TKT yang digunakan sebagai acuan kelas topik final.
5. Memperkaya corpus dapat menghasilkan hasil yang baik dengan cara analisis manual terhadap hasil memperkaya corpus dengan POS Tagging agar mendapatkan kelas kata yang tepat dan seimbang.
6. Penggunaan data berbasis SINTA dalam Tugas Akhir ini kurang valid karena data yang digunakan tidak diperbarui secara berkala oleh pihak SINTA.
7. Kombinasi kriteria pemeringkatan yang bersifat kualitatif dan kuantitatif dapat menghasilkan hasil pemeringkatan yang mendekati Ground Truth sistem yaitu QS University Rankings 2016-2017.

6.2 Saran

Beberapa saran yang dapat diberikan untuk pengembangan sistem lebih lanjut adalah sebagai berikut:

1. Memperkaya data kumpulan kata-kata dalam masing-masing TKT untuk memberikan hasil akurasi klasifikasi yang lebih optimal.
2. Memperkaya kriteria/indikator metode penilaian pemeringkatan universitas untuk meningkatkan akurasi dalam perbandingan dengan peringkat QS.
3. Melakukan analisis secara manual terhadap dokumen corpus hasil POS Tagging untuk memberikan hasil akurasi yang lebih optimal.
4. Memperkaya kriteria/indikator metode penilaian pemeringkatan universitas untuk memberikan opsi pembobotan *custom* yang lebih banyak bagi pengguna situs.
5. Pengembangan perangkat lunak kedepannya disarankan menggunakan data untuk kriteria Jumlah Sitasi dan kualitas Sitasi maupun data yang bersifat kuantitatif terhadap akademisi menggunakan data dari Scopus, karena data yang disediakan telah lengkap dan diperbarui secara berkala.

DAFTAR PUSTAKA

- Addison-Wesley. (2013). Learn Python the Hard Way: A Very Simple Introduction. In Z. A. Shaw, *Learn Python the Hard Way: A Very Simple Introduction* (pp. 14-89). Boston: Addison-Wesley.
- Adina - Petruta Pavel. (2015). Global University Rankings - A Comparative Analysis. *Procedia Economics and Finance* 26, 54-63.
- Andreyan Rizky Baskara, Riyanarto Sarno, Adhatus Solichah. (2016). Discovering Traceability between Business Process and Software Component using Latent Dirichlet Allocation.
- Anne-Wil K. Harzing, Ron van der Wal. (2008). Google Scholar as A New Source for Citation Analysis. *Ethics in Science and Environmental Politics*, 61-73.
- Arjun Thakur, A. L. Sangal, Harminder Bindra. (2011). Quantitative Measurement and Comparison of Effects of Various SearchEngine Optimization Parameters on Alexa Traffic Rank. *International Journal of Computer Applications*, 975-8887.
- Badan Akreditasi Nasional Perguruan Tinggi. (2011). *Pedoman Penyusunan Borang Akreditasi Institusi Perguruan Tinggi*. Jakarta: Badan Akreditasi Nasional Perguruan Tinggi.
- Bassam Al-Salemi, M. J. (2015). Journal of Information Science. *LDA-AdaBoost.MH: Accelerated AdaBoost.MH based on latent Dirichlet allocation for text categorization*, 27-40.
- Bassam Al-Salemi, M. J. (2015). Journal of Information Science. *Boosting algorithms with topic modeling for multi-label text categorization: A comparative empirical study*, 732-746.
- Center of Teaching and Learning Stanford University. (1981). *Bloom's Taxonomy of Educational Objectives*. California: Stanford University.

- Cornell University Library. (2017). *Cornell University Library*. Retrieved from Measuring your research impact: i10-Index: <http://guides.library.cornell.edu/c.php?g=32272&p=203393>
- Daniel Jurafsky, James H. Martin. (2016). Part-of-Speech Tagging. In J. H. Daniel Jurafsky, *Speech and Language Processing*.
- Daniel Ramage, David Hall, Ramesh Nallapati, Christopher D. Manning. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora.
- David M. Blei, Andrew Y. Ng, Michael I. Jordan. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research* 20, 993-1022.
- David M. Blei, Jon D. McAuliffe. (2007). Supervised Topic Model. *Advances in Neural Information Processing Systems*.
- Dorough, B. (2014). Part-of-Speech Tagging. In D. J. Martin, *Speech and Language Processing* (pp. 1-28). New York City: Pearson Education.
- Downing, K. G. (2016). World University Rankings and the Future of Higher Education. In K. G. Downing, *World University Rankings and the Future of Higher Education* (p. 214). Hong Kong: IGI Global. Retrieved from World Universities Search Engine: <http://www.4icu.org/about/>
- Downing, K. G. (2016). World University Rankings and the Future of Higher Education. In K. G. Downing, *World University Rankings and the Future of Higher Education* (p. 213). Hong Kong: IGI Global.
- Elsevier. (2013). *About Scopus*. Retrieved from Elsevier: <https://www.elsevier.com/solutions/scopus>
- Elsevier. (2016, January 1). *Elsevier Developers*. Retrieved December 10, 2016, from <http://dev.elsevier.com/>
- ementarian Riset Teknologi Dan Pendidikan Tinggi Republik Indonesia. (2017). *SINTA*. Retrieved from SINTA (Science and Technology Index): <http://sinta1.ristekdikti.go.id/?ref=about>

- Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 2825-2830.
- John Wiley & Sons. (2011). HTML and CSS: Design and Build Websites. In J. Duckett, *HTML and CSS: Design and Build Websites* (pp. 13-39). New Jersey: John Wiley & Sons.
- Juan Ramos. (2003). Using TF-IDF to Determine Word Relevance in Document Queries.
- K.S. Reddy, En Xie, Qingqing Tang. (2016). Higher education, high-impact research, and world university rankings: A Case of India and comparison with China. *Pacific Science Review B: Humanities and Social Sciences* 2, 1-21.
- Kementerian Riset Teknologi dan Pendidikan Tinggi Republik Indonesia. (2016, January 1). *Kementerian Riset Teknologi dan Pendidikan Tinggi Republik Indonesia*. Retrieved December 10, 2016, from <http://www.forlap.dikti.go.id/>
- Kementerian Riset, Teknologi dan Pendidikan Tinggi. (2017). *Profil, Fitur, Registrasi Sinta Pusat Index, Sitasi dan Kepakaran Terbesar di Indonesia*. Jakarta: Kementerian Riset, Teknologi dan Pendidikan Tinggi.
- Kementerian Riset, teknologi dan Pendidikan Tinggi. (2017, July 28). *Sinta Science and Technology Index*. (Kementerian Riset Teknologi dan Pendidikan Tinggi) Retrieved July 28, 2017, from <http://sinta1.ristekdikti.go.id/>
- Kementerian, Riset Teknologi dan Pendidikan Tinggi. (2016). *Peraturan Menteri Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia Nomor 42 Tahun 2016 Tentang Pengukuran dan Penetapan Tingkat Kesiapterapan Teknologi*. Jakarta: Kementerian Riset, Teknologi dan Pendidikan Tinggi.
- Majelis Akreditasi, Badan Akreditasi Nasional Perguruan Tinggi Indonesia. (2017). *Kebijakan Penyusunan Instrumen Akreditasi*. Jakarta: Majelis Akreditasi, Badan Akreditasi Nasional Perguruan Tinggi Indonesia.

- Menteri Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia. (2016). *Pengukuran dan Penetapan Tingkat Kesiapterapan Teknologi*. Jakarta: Menteri Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia.
- Mu-Hsuan Huang. (2012). Opening The Black Box of QS World University Rankings. *Research Evaluation* 21, 71-78.
- Nikhil Johri, Daniel Ramage, Daniel A. McFarland, Daniel Jurafsky. (2011). A Study of Academic Collaboration in Computational Linguistics with Latent Mixtures of Authors. *LaTeCH '11 Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, 124-132.
- O'Reilly Media, Inc. (2004). Web Database Applications with PHP and MySQL. In D. L. Hugh E. Williams, *Web Database Applications with PHP and MySQL: Building Effective Database* (pp. 16-53, 135-168). Sebastopol, California: O'Reilly Media, Inc.
- Pavlina, K. (2012). Webometric Ranking of European Universities. *Procedia - Social and Behavioral Sciences*, 3788-3792.
- Pusat Data dan Informasi Iptek Dikti. (2016). *Pangkalan Data Pendidikan Tinggi (PDDIKTI)*. Jakarta: Pusat Data dan Informasi Iptek Dikti.
- QS World University Rankings. (2014). *Annual QS Global Employer Survey*. QS World University Rankings.
- QS World University Rankings. (2014). *QS Global Academic Survey 2014*. QS World University Rankings.
- QS World University Rankings. (2016, January 1). *QS Top University*. Retrieved December 10, 2016, from <http://www.topuniversities.com/university-rankings/asian-university-rankings/2016>
- Raymond J. Mooney, L. R. (2000). Content-based book recommending using learning for text categorization. *DL '00 Proceedings of the fifth ACM conference on Digital libraries*. San Antonio, Texas, USA.
- Simon Lacoste-Julien, Fei Sha, Michael I. Jordan. (2008). DiscLDA Discriminative Learning for Dimensionality

- Reduction and Classification. *Advances in Neural Information Processing Systems* 21.
- Steven Bird. (2006). NLTK: The Natural Language Toolkit. *Proceedings of the COLINGACL on Interactive presentation*, 69-72.
- Tatiana Sidorenko, Tatiana Gorbatova. (2015). Efficiency of Russian education through the scale of. *Procedia - Social and Behavioral Sciences* 166, 464-467.
- TextBlob. (2017, June 28). *TextBlob: Simplified Text Processing*. Retrieved June 28, 2017, from <https://textblob.readthedocs.io/en/dev/>
- Tingting Wei, Yonghe Lu, Houyou Chang, Qiang Zhou, Xianyu Bao. (2015). A Semantic Approach for Text Clustering using WordNet and Lexical Chains. *Expert Systems with Applications* 42, 4(1), 2264-2275.
- Vidya Rajiv Yeravdekar, Gauri Tiwari. (2014). Global Rankings of Higher Education Institutions and India's Effective Non-Precense: Why Have World-Class Universities Eluded The Indian Higher Education System? And, How Worthwhile is The Indian Government's Captivation to Launch World Class Universities. *Procedia - Social and Behavioral Sciences* 157, 63-83.
- Wikipedia. (2004). *QS World University Rankings*. Retrieved from Wikipedia:
https://en.wikipedia.org/wiki/QS_World_University_Rankings
- Wikipedia. (2016, December 14). *Wikipedia College and University Rankings*. Retrieved December 18, 2016, from https://en.wikipedia.org/wiki/College_and_university_rankings
- Wikipedia. (2016, August 4). *Wikipedia Scopus*. Retrieved December 10, 2016, from <https://en.wikipedia.org/wiki/Scopus>
- Wikipedia. (2017). *Google Scholar*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Google_Scholar

- Wikipedia. (2017). *h-index*. Retrieved from Wikipedia: <https://en.wikipedia.org/wiki/H-index>
- Wikipedia. (2017). *tf-idf*. Retrieved from Wikipedia: <https://en.wikipedia.org/wiki/Tf%E2%80%93idf>
- Wikipedia. (2017, June 27). *Wikipedia Google Scholar*. Retrieved December 17, 2016, from https://en.wikipedia.org/wiki/Google_Scholar
- Wikipedia. (2017, July 28). *Wikipedia H-Index*. Retrieved July 28, 2017, from <https://en.wikipedia.org/wiki/H-index>
- Wikipedia. (2017, July 28). *Wikipedia MySQL*. Retrieved July 28, 2017, from <https://en.wikipedia.org/wiki/MySQL>
- Wikipedia. (2017, July 28). *Wikipedia PHP*. Retrieved July 28, 2017, from <https://en.wikipedia.org/wiki/PHP>
- Wikipedia. (2017, July 28). *Wikipedia Python Programming Language*. Retrieved July 28, 2017, from [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- William M. Darling. (2011, December 1). A Theoretical and Practical Implementation Tutorial on Topic Modeling and Gibbs Sampling.
- Zulhanif. (2016). Pemodelan Topik dengan Latent Dirichlet Allocation.

LAMPIRAN

Kode Sumber

```
#!/usr/bin/python
# -*- coding: latin-1 -*-

import sys
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem import SnowballStemmer
from nltk.tokenize import word_tokenize, sent_tokenize,
RegexpTokenizer
import re, string, nltk, os, math, random, time
from textblob import TextBlob

#fungsi sorting str+number (campuran) tipe karakter
def atoi(text):
    return int(text) if text.isdigit() else text

def natural_keys(text):
    return [ atoi(c) for c in re.split('(\d+)', text) ]

# def perkaya_corpus_tkt(tmp_result, list_update_tkt_all,
list_update_tkt2, list_update_tkt3, list_update_tkt4,
list_update_tkt5, list_update_tkt6, list_update_tkt7,
list_update_tkt8, list_update_tkt9, list_update_tkt_all):
def perkaya_corpus_tkt(tmp_result, list_update_tkt_all):
    openfile_tkt1 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
RU_UPDATE/dokumen_tkt/A_Datatrain/TKT1.txt", "r")
    content_tkt1 = openfile_tkt1.read()
    content_tkt1 = content_tkt1.split()
```

```

    openfile_tkt1.close()
    # list_update_tkt_all.append(content_tkt1)

    openfile_tkt2 =
    open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
    RU_UPDATE/dokumen_tkt/A_Datatrain/TKT2.txt", "r")
    content_tkt2 = openfile_tkt2.read()
    content_tkt2 = content_tkt2.split()
    openfile_tkt2.close()

    openfile_tkt3 =
    open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
    RU_UPDATE/dokumen_tkt/A_Datatrain/TKT3.txt", "r")
    content_tkt3 = openfile_tkt3.read()
    content_tkt3 = content_tkt3.split()
    openfile_tkt3.close()

    openfile_tkt4 =
    open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
    RU_UPDATE/dokumen_tkt/A_Datatrain/TKT4.txt", "r")
    content_tkt4 = openfile_tkt4.read()
    content_tkt4 = content_tkt4.split()
    openfile_tkt4.close()

    openfile_tkt5 =
    open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
    RU_UPDATE/dokumen_tkt/A_Datatrain/TKT5.txt", "r")
    content_tkt5 = openfile_tkt5.read()
    content_tkt5 = content_tkt5.split()
    openfile_tkt5.close()

    openfile_tkt6 =
    open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
    RU_UPDATE/dokumen_tkt/A_Datatrain/TKT6.txt", "r")
    content_tkt6 = openfile_tkt6.read()
    content_tkt6 = content_tkt6.split()

```



```

    openfile_tkt6.close()

    openfile_tkt7                                =
    open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
    RU_UPDATE/dokumen_tkt/A_Datatrain/TKT7.txt", "r")
    content_tkt7 = openfile_tkt7.read()
    content_tkt7 = content_tkt7.split()
    openfile_tkt7.close()

    openfile_tkt8                                =
    open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
    RU_UPDATE/dokumen_tkt/A_Datatrain/TKT8.txt", "r")
    content_tkt8 = openfile_tkt8.read()
    content_tkt8 = content_tkt8.split()
    openfile_tkt8.close()

    openfile_tkt9                                =
    open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
    RU_UPDATE/dokumen_tkt/A_Datatrain/TKT9.txt", "r")
    content_tkt9 = openfile_tkt9.read()
    content_tkt9 = content_tkt9.split()
    openfile_tkt9.close()

    tmp_result = tmp_result.split()
    tmp_word = ""
    listWord = []
    lw = { }
    tmp_result_2 = ""

    for word in tmp_result:
        if word in lw:
            lw[word] += 1;
        else:
            lw[word] = 1;
            tmp_result_2 += word + " "

```

```

listWord.append(lw)
tmp_result_2_final = tmp_result_2.split()

print tmp_result_2_final

#tambahan
index=0
for word in tmp_result_2_final:

    if word in content_tkt1:
        text_postagger = nltk.word_tokenize(word)
        pos = nltk.pos_tag(text_postagger)
        list_update_tkt_all.append(word)

        for i in range(0, len(pos)):
            jos = pos[i][1]
            if jos=='NN':

#menentukan asumsi label awal untuk inputan komputasi
#fungsi untuk menentukan label awal menggunakan keyword
bloom taxonomy (fix)
def set_label(tmp_result):
    sys.stderr.write("JUMLAH KATA DALAM DOKUMEN:
"+str(len(tmp_result))+"\n")

    openfile_tkt1 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
RU_UPDATE/dokumen_tkt/A_Datatraining_Update/update_TKT
1.txt", "r")
    content_tkt1 = openfile_tkt1.read()
    openfile_tkt1.close()

    openfile_tkt2 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA

```

```

RU_UPDATE/dokumen_tkt/A_Datatrain_Update/update_TKT
2.txt", "r")
    content_tkt2 = openfile_tkt2.read()
    openfile_tkt2.close()

    openfile_tkt3                                =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
RU_UPDATE/dokumen_tkt/A_Datatrain_Update/update_TKT
3.txt", "r")
    content_tkt3 = openfile_tkt3.read()
    openfile_tkt3.close()

    openfile_tkt4                                =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
RU_UPDATE/dokumen_tkt/A_Datatrain_Update/update_TKT
4.txt", "r")
    content_tkt4 = openfile_tkt4.read()
    openfile_tkt4.close()

    openfile_tkt5                                =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
RU_UPDATE/dokumen_tkt/A_Datatrain_Update/update_TKT
5.txt", "r")
    content_tkt5 = openfile_tkt5.read()
    openfile_tkt5.close()

    openfile_tkt6                                =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
RU_UPDATE/dokumen_tkt/A_Datatrain_Update/update_TKT
6.txt", "r")
    content_tkt6 = openfile_tkt6.read()
    openfile_tkt6.close()

    openfile_tkt7                                =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA

```

```

RU_UPDATE/dokumen_tkt/A_Datatrain_Update/update_TKT
7.txt", "r")
    content_tkt7 = openfile_tkt7.read()
    openfile_tkt7.close()

    openfile_tkt8
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
RU_UPDATE/dokumen_tkt/A_Datatrain_Update/update_TKT
8.txt", "r")
    content_tkt8 = openfile_tkt8.read()
    openfile_tkt8.close()

    openfile_tkt9
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
RU_UPDATE/dokumen_tkt/A_Datatrain_Update/update_TKT
9.txt", "r")
    content_tkt9 = openfile_tkt9.read()
    openfile_tkt9.close()

#UPDATE
    openfile_tkt_all
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
RU_UPDATE/dokumen_tkt/A_Datatrain_All/tampung_TKT.t
xt", "r")
    content_tkt_all = openfile_tkt_all.read()
    openfile_tkt_all.close()

counter1=counter2=counter3=counter4=counter5=counter6=co
unter7=counter8=counter9=counter_all=0

tmp_result = tmp_result.split()
tmp_word = ""

#hitung frekuensi kemunculan kata
listWord = []

```

```
lw = {}
tmp_result_2 = ""

for word in tmp_result:
    if word in lw:
        lw[word] += 1;
    else:
        lw[word] = 1;
        tmp_result_2 += word + " "

listWord.append(lw)

tmp_result_2_final = tmp_result_2.split()

for word in tmp_result_2_final:
    if word in content_tkt1:
        counter1 += 1

    if word in content_tkt2:
        counter2 += 1

    if word in content_tkt3:
        counter3 += 1

    if word in content_tkt4:
        counter4 += 1

    if word in content_tkt5:
        counter5 += 1

    if word in content_tkt6:
        counter6 += 1

    if word in content_tkt7:
        counter7 += 1
```

```

if word in content_tkt8:
    counter8 += 1

if word in content_tkt9:
    counter9 += 1

list_label = [counter1, counter2, counter3, counter4, counter5,
counter6, counter7, counter8, counter9]
list_nama_label =
["T1", "T2", "T3", "T4", "T5", "T6", "T7", "T8", "T9"]

tmp_lnl = { }
a=0
for lnl in list_nama_label:
    tmp_lnl[a] = lnl
    a+=1

tmp_lb = { }
b=0
p = ""
simpan = []
for lb in list_label:
    if lb>0:
        tmp_lb[b] = lb
        simpan.append(str(tmp_lb[b])+"-"+str(tmp_lnl[b]))
        p += tmp_lnl[b] + ","

    b+=1

a = sorted(simpan, key=natural_keys, reverse=True)
sys.stderr.write("SORTING HASIL TF (pencocokan
berdasarkan term): "+str(a)+"\n")

ambil_4_label_teratas = a[:4]

```

```

ambil_4_label_teratas = ''.join(ambil_4_label_teratas)

b = ambil_4_label_teratas.split()

c = {}
d=0
e=0
f = {}
label_final = ""
for j in b:
    d+=1
    # c[d] = j[2:]
    c[d] = j[-2:]
    # sys.stderr.write("NOPE:"+c[d]+"\\n")

for k in b:
    label_final += str(c[d-e])+", "
    e+=1

label_final = label_final[:-1]
# sys.stderr.write("POMPOM:"+label_final+"\\n")
label_final_akhir = "["+label_final+"]"
# sys.stderr.write("POMPOM:"+label_final_akhir+"\\n")
label_final_akhir = re.sub('-', ' ', label_final_akhir)
# sys.stderr.write("POMPOM:"+label_final_akhir+"\\n")

return label_final_akhir

#fungsi untuk preprocessing teks bhs inggris menggunakan
snowball stemmer, nltk stopword, remove punctuation (fix)
def inggris(dir3, nama_universitas, dir2_input, lala2, listTF,
tipe_bahasa,          list_update_tkt1,          list_update_tkt2,
list_update_tkt3,          list_update_tkt4,          list_update_tkt5,
list_update_tkt6,          list_update_tkt7,          list_update_tkt8,
list_update_tkt9, list_update_tkt_all):
    tipe_bahasa = " - TIPE INGGRIS"

```

```

nama_file = "Dokumen_"

# 1 tokenisasi & remove punctuation
file_input = dir2_input.lower().split()

punctuation = re.compile(r'^\w.'])
word_list_punctuation = [punctuation.sub(" ", word) for word
in file_input]
tmp_word_list_punctuation = [x.strip(' ') for x in
word_list_punctuation]

punctuation2 = re.compile(r'[-.?!,":;()|0-9]')
word_list_punctuation2 = [punctuation2.sub("", word) for
word in tmp_word_list_punctuation]
tmp_word_list_punctuation2 = [x.strip(' ') for x in
word_list_punctuation2]

# 2 stoptords & stemming
snowball_stemmer = SnowballStemmer("english")
stop_words = {}
stop_words = set(stopwords.words("english"))

words_list = []
for i in tmp_word_list_punctuation2:
    if i not in stop_words:
        words_list.append(i)

listResult = []
for word in words_list:
    listResult.append(word)

```



```

tf = {}
for word in listResult:
    if word in tf:
        tf[word] += 1;
    else:
        tf[word] = 1;

listTF.append(tf)

tmp_result = ""

for word in listResult:
    if len(word) > 2:
        tmp_result += word + " "

jumlah_kata_hasil_preprocessing = tmp_result.split()
sys.stderr.write("JUMLAH          KATA          HASIL
PREPROCESSING:
"+str(len(jumlah_kata_hasil_preprocessing))+ "\n")
dir3.write("JUMLAH KATA HASIL PREPROCESSING:
"+str(len(jumlah_kata_hasil_preprocessing))+ "\n\n")

path_full_doc =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_U
PDATE/result_preprocessing_dan_labeling_bloom/"+nama_un
iversitas+"/amat_final_full_result_preprocessing_"
nama_file_full_doc_result = path_full_doc+lala2+".txt"
final_full = open(nama_file_full_doc_result, 'w')

final_full.write(tmp_result)
final_full.close() #=====
print hasil preprocessing (bersih) dari satu file dokumen yg
isinya banyak =====

```

```

#UPDATE
perkaya_corpus_tkt(tmp_result, list_update_tkt_all)

#UPDATE
path_update_tkt_all =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_U
PDATE/dokumen_tkt/A_Datatrain_All/tampung_TKT_"+nam
a_universitas+".txt"
final_tkt_all = open(path_update_tkt_all, 'w')

for i in range(0, len(list_update_tkt_all)):
    final_tkt_all.write(list_update_tkt_all[i]+"\\n")      #
===== write kata2 hasil perkara corpus ditaruh dalam
satu file baru bernama tampung =====
    sys.stderr.write(list_update_tkt_all[i]+"\\n")

final_tkt_all.close()

# sys.stderr.write(set_label(tmp_result)+" "+tmp_result)
# print set_label(tmp_result), tmp_result                #
=====          IDAK          PAKAI          INI
=====

# sys.stderr.write(tmp_result)
sys.stderr.write("-- Not Important!!! --\\n")
sys.stderr.write("SUKSES PREPROCESS!!!\\n\\n")

#fungsi untuk translate dokumen bhs indo ke bhs inggris (fix)
def translate_to_english_function(inputan_file, nFiles, listTF):
    isi_file = inputan_file
    jumlah_file = nFiles

```

```

list_tf = listTF
tipe_bahasa = " - TIPE INDONESIA"

en_blob = TextBlob(isi_file)
hasil_translate = en_blob.translate(to='en')
sys.stderr.write("HASIL
TRANSLATE:"+tipe_bahasa+"\n"+str(hasil_translate)+"\n\n")

inggris(hasil_translate, jumlah_file, list_tf, tipe_bahasa)

#fungsi persiapan awal (fix)
def compute():
    nama_universitas = "Universitas_Sebelas_Maret"
    path =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_U
PDATE/dokumen_universitas/"+nama_universitas+"/"
    dirs = os.listdir(path)
    lala = path[71:]
    lala2 = lala[:-1]
    print lala2 #===== read all
dokumen universitas =====

#tambahan
path_full_doc =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_U
PDATE/result_preprocessing_dan_labeling_bloom/"+nama_un
iversitas+"/full_result_preprocessing_"
    ekstensi_full_doc = ".txt"
    nama_file_full_doc_result =
path_full_doc+lala2+ekstensi_full_doc

```

```

    final      =      open(nama_file_full_doc_result,      'w')
#===== read all dokumen universitas kemudian di jadikan
satu file (belum bersih) =====

```

```

nFiles = 0
listTF = []
tipe_bahasa = ""

```

```

list_update_tkt1 = []
list_update_tkt2 = []
list_update_tkt3 = []
list_update_tkt4 = []
list_update_tkt5 = []
list_update_tkt6 = []
list_update_tkt7 = []
list_update_tkt8 = []
list_update_tkt9 = []

```

```

#UPDATE
list_update_tkt_all = []

```

```

for eachFile in dirs:
    nFiles += 1
    with open(path+eachFile, 'r+') as myFile:
        inputan_file = myFile.read().replace('\n', ' ')
        #      sys.stderr.write("Data      ke      "+str(nFiles)+":
"+str(inputan_file)+"\n\n")
        final.write(inputan_file+" ")
    final.close()

```

```

    dir2
    open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA

```

```

RU_UPDATE/result_preprocessing_dan_labeling_bloom/"+na
ma_universitas+"/full_result_preprocessing_"+lala2+".txt", "r")
    dir2_input = dir2.read() #===== read
dokumen yg full (belum bersih) untuk di preprocessing
=====
    # sys.stderr.write(dir2_input)

    dir3 =
open("C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BA
RU_UPDATE/result_preprocessing_dan_labeling_bloom/"+na
ma_universitas+"/analisa_hasil_preprocessing_"+lala2+".txt",
"w")
    jumlah_kata = dir2_input.split()
    sys.stderr.write("JUMLAH          SEMUA          KATA:
"+str(len(jumlah_kata))+ "\n")
    dir3.write("JUMLAH          SEMUA          KATA:
"+str(len(jumlah_kata))+ "\n\n")

    inggris(dir3, nama_universitas, dir2_input, lala2, listTF,
tipe_bahasa, list_update_tkt1, list_update_tkt2,
list_update_tkt3, list_update_tkt4, list_update_tkt5,
list_update_tkt6, list_update_tkt7, list_update_tkt8,
list_update_tkt9, list_update_tkt_all)

    dir3.close()

#MAIN FUNCTION#
if __name__ == "__main__":
    start_time = time.time()

    compute()

    time_akhir = time.time() - start_time

```

```
sys.stderr.write("--- "+str(time_akhir)+" detik ---")
```

Kode Sumber A.1 Preprosesing

```
import os
import nltk
import tfidf
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer

pathroot='C:/Users/USER/Downloads/arga/dokumen_universit
as'
pathtrain = 'C:/Users/USER/Downloads/arga/A_Datatraining'
pathsimi='C:/Users/USER/Downloads/arga/dokumen_universit
as_merge'
train = []

#----- BACA INPUT DAN APDET TKT -
-----#
for foldername in os.listdir(pathroot):
    print 'proses ' + foldername
    pathnya=pathroot + '/' + foldername
    simpan=pathroot+ '_merge/' + foldername + '_simpan.txt'
    f=open(simpan,'w')
    for filename in os.listdir(pathnya):
        filename=pathnya+'/'+filename
        ff=open(filename, 'r')
        content=ff.read()
        f.write(content.lower())
        ff.close()

f=open(simpan,'r')
ditag=f.read()
ditag=ditag.replace("-", " ")
```

```

ditag=ditag.translate(None, string.punctuation)
ditag=ditag.decode("utf8", 'ignore')
token_ditag=nltk.word_tokenize(ditag)
ditag=nltk.pos_tag(token_ditag)

```

```

hasile_postag=[]
next_postag=[]
index=0
for kata, jenis in ditag:
    if 'V' in jenis:
        if kata not in hasile_postag:
            hasile_postag.append(kata)
            if index==0:
                next_a, next_b=ditag[index+1]
                tupel=(kata, next_a)
                next_postag.append(tupel)
            elif index == len(ditag)-1:
                prev_a, prev_b = ditag[index - 1]
                tupel = (prev_a, kata)
                next_postag.append(tupel)
            else:
                next_a, next_b=ditag[index+1]
                prev_a, prev_b = ditag[index - 1]
                tupel=(prev_a, kata, next_a)
                next_postag.append(tupel)

```

```

index=index+1

```

```

f.close()

```

```

fname=[]
for filename in os.listdir(pathtrain):
    filename2=pathtrain+'new/'+filename
    fname.append(filename2)
    filename=pathtrain+'/'+filename
    ff=open(filename, 'r')

```

```

content=ff.read()
content = content.decode("utf8", 'ignore')
content = nltk.word_tokenize(content)
temp=[]
for kata in content:
    temp.append(kata)
train.append(temp)
ff.close()

indexkata=0
for kata in hasile_postag:
    index=0
    for lis in train:
        if kata in lis:
            # print kata
            # print fname[index]
            # print next_postag[indexkata]
            for ditambah in next_postag[indexkata]:
                if ditambah not in lis:
                    # print ditambah
                    lis.append(ditambah)

            index=index+1
            indexkata=indexkata+1

for i in range (0,9):
    # # print i
    # print fname[i]
    f=open(fname[i], 'w')
    for kata in train[i]:
        f.write(kata)
        f.write("\n")
    f.close
#----- BACA INPUT DAN APDET TKT -
-----#

```

Kode Sumber A.2 Pembaruan Korpus


```

#----- TF IDF -----
-#
def getKey(item):
    return item[1]

table = tfidf.TfIdf()
table.add_document("tk1", train[0])
table.add_document("tk2", train[1])
table.add_document("tk3", train[2])
table.add_document("tk4", train[3])
table.add_document("tk5", train[4])
table.add_document("tk6", train[5])
table.add_document("tk7", train[6])
table.add_document("tk8", train[7])
table.add_document("tk9", train[8])

hasil_akhir = [['a' for x in range(3)] for y in range(17)]
nilai_hasil_akhir = [[0 for x in range(3)] for y in range(17)]

index=0
for filename in os.listdir(pathsimi):
    print 'menghitung tf idf ' + filename
    f = open(pathsimi+'/'+filename, 'r')
    ditag = f.read()
    ditag = ditag.replace("-", " ")
    ditag = ditag.translate(None, string.punctuation)
    ditag = ditag.decode("utf8", 'ignore')
    token_ditag = nltk.word_tokenize(ditag)
    hasil_simi=table.similarities(token_ditag)
    hasil_simi=sorted(hasil_simi,key=getKey, reverse=True)
    for i in range(3):
        hasil_akhir[index][i]=hasil_simi[i][0]
        nilai_hasil_akhir[index][i] = hasil_simi[i][1]
    index=index+1

```

```

hasile=open('hasil tf idf.txt', 'w')
index=0

for foldername in os.listdir(pathroot):
    hasile.write(foldername)
    hasile.write('\n')
    for tulis in hasil_akhir[index]:
        hasile.write(str(tulis))
        hasile.write(' ')
    hasile.write('\n')
    for tulis in nilai_hasil_akhir[index]:
        hasile.write(str(tulis))
        hasile.write(' ')
    hasile.write('\n\n')
    index=index+1
hasile.close()
#----- TF IDF -----#

```

Kode Sumber A.3 Klasifikasi TF

```

ps = PorterStemmer()
for filename in os.listdir(pathsimi):
    print 'menghitung LDA ' + filename
    f = open(pathsimi+'/'+filename, 'r')
    ditag = f.read()
    ditag = ditag.replace("-", " ")
    ditag = ditag.translate(None, string.punctuation)
    ditag = ditag.decode("utf8", 'ignore')
    token_ditag = nltk.word_tokenize(ditag)
    stopWords = set(stopwords.words('english'))
    ditag_clean=[]
    for kata in token_ditag:
        if kata not in stopWords:
            ditag_clean.append(ps.stem(kata))

import numpy as np
import matplotlib.pyplot as plt

```

```

from sklearn.tree import DecisionTreeClassifier
from rf_classification import get_data

class AdaBoost:
    def __init__(self, M):
        self.M = M

    def fit(self, X, Y):
        self.models = []
        self.alphas = []

        N, _ = X.shape
        W = np.ones(N) / N

        for m in xrange(self.M):
            tree = DecisionTreeClassifier(max_depth=1)
            tree.fit(X, Y, sample_weight=W)
            P = tree.predict(X)

            err = W.dot(P != Y)
            alpha = 0.5*(np.log(1 - err) - np.log(err))

            W = W*np.exp(-alpha*Y*P) # vectorized form
            W = W / W.sum() # normalize so it sums to 1

            self.models.append(tree)
            self.alphas.append(alpha)

    def predict(self, X):
        # NOT like SKLearn API
        # we want accuracy and exponential loss for plotting purposes
        N, _ = X.shape
        FX = np.zeros(N)
        for alpha, tree in zip(self.alphas, self.models):
            FX += alpha*tree.predict(X)

```

```

    return np.sign(FX), FX

def score(self, X, Y):
    # NOT like SKLearn API
    # we want accuracy and exponential loss for plotting purposes
    P, FX = self.predict(X)
    L = np.exp(-Y*FX).mean()
    return np.mean(P == Y), L

if __name__ == '__main__':

    X, Y = get_data()
    Y[Y == 0] = -1 # make the targets -1,+1
    Ntrain = int(0.8*len(X))
    Xtrain, Ytrain = X[:Ntrain], Y[:Ntrain]
    Xtest, Ytest = X[Ntrain:], Y[Ntrain:]

    T = 200
    train_errors = np.empty(T)
    test_losses = np.empty(T)
    test_errors = np.empty(T)
    for num_trees in xrange(T):
        if num_trees == 0:
            train_errors[num_trees] = None
            test_errors[num_trees] = None
            test_losses[num_trees] = None
            continue
        if num_trees % 20 == 0:
            print num_trees

        model = AdaBoost(num_trees)
        model.fit(Xtrain, Ytrain)
        acc, loss = model.score(Xtest, Ytest)
        acc_train, _ = model.score(Xtrain, Ytrain)
        train_errors[num_trees] = 1 - acc_train

```

```
test_errors[num_trees] = 1 - acc
test_losses[num_trees] = loss

if num_trees == T - 1:
    print "final train error:", 1 - acc_train
    print "final test error:", 1 - acc

plt.plot(test_errors, label='test errors')
plt.plot(test_losses, label='test losses')
plt.legend()
plt.show()

plt.plot(train_errors, label='train errors')
plt.plot(test_errors, label='test errors')
plt.legend()
plt.show()
```

```

#-----AdaBoost-----#
#!/usr/bin/python
# -*- coding: latin-1 -*-

from __future__ import division

import sys
# from nltk.corpus import stopwords
# from nltk.stem import PorterStemmer
# from nltk.stem import SnowballStemmer
# from nltk.tokenize import word_tokenize, sent_tokenize,
RegexpTokenizer
import re, string, nltk, os, math, random, time
# from textblob import TextBlob
from os import listdir
from os.path import isfile, join

path =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_U
PDATE/result_lda_with_tf_idf_sklearn/"
onlyfiles = [f for f in listdir(path) if isfile(join(path, f))]

nama_file = "result_theta_max_"
list_label = {}
list_label_2 = []

for i in range(0, len(onlyfiles)):
    if nama_file in onlyfiles[i]:
        # print onlyfiles[i]
        dir2 = open(path+onlyfiles[i], "r").read()
        # print dir2.split()
        dom = dir2.split()

        for label in dom:
            if label in list_label:

```

```

                                list_label[label]+=1
                    else:
                                list_label[label]=1

list_label_2.append(list_label)
# print list_label_2

# for key, value in list_label_2[0].iteritems():
#     print key
#     print value

nama_file = "result_theta_all_Institut Pertanian Bogor 9.txt"
fname =
"C:/xampp/htdocs/TA/1_REPUTASI_AKADEMIK_BARU_U
PDATE/result_lda_with_tf_idf_sklearn/"+nama_file

lala = fname[100:]
nama_universitas = lala[:-4]
print nama_universitas

with open(fname) as f:
    content = f.readlines()

content = [x.strip() for x in content]
simpan_label = { }
simpan_probs = { }
jumlah_label=0
tmp_jumlah_label_tertinggi = 0

for i in range(0, len(content)):
    simpan_label[i] = content[i][:2]
    simpan_probs[i] = content[i][3:]

    print simpan_label[i], simpan_probs[i]

```

```

        for key, value in list_label_2[0].iteritems():
            if simpan_label[i] in key:
                print simpan_label[i], "sama", key, value
                tmp_jumlah_label_tertinggi += value

print          "JUMLAH          TOTAL          LABEL",
tmp_jumlah_label_tertinggi, "\n"

total_caca = 0

for i in range(0, len(content)):
    simpan_label[i] = content[i][:2]
    simpan_probs[i] = content[i][3:]

    for key, value in list_label_2[0].iteritems():

        if simpan_label[i] in key:
            adabost = value/tmp_jumlah_label_tertinggi
            bobo = (float(simpan_probs[i])+adabost)/2

            coco = re.findall(r'\d+', simpan_label[i])
            caca = bobo*float(coco[0])

            print key, adabost, bobo, caca

            total_caca+=caca

hasil_akhir = total_caca/9
print "HASIL AKHIR", hasil_akhir

```

Kode Sumber A.4 LDA-AdaBoost.MH


```

CREATE DATABASE /*!32312 IF NOT
EXISTS*/`database_qs_2` /*!40100 DEFAULT CHARACTER
SET latin1 */;

USE `database_qs_2`;

CREATE TABLE `universitas` (
  `id` int(2) unsigned NOT NULL AUTO_INCREMENT,
  `nama` varchar(255) NOT NULL,
  `domain` varchar(150) NOT NULL,
  `url_web` varchar(150) NOT NULL,
  `kota` varchar(150) NOT NULL,
  `provinsi` varchar(150) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=410 DEFAULT
CHARSET=latin1;
CREATE TABLE `reputasi_akademik` (
  `id` int(2) unsigned NOT NULL
AUTO_INCREMENT,
  `nama_universitas` varchar(255) NOT NULL,
  `skor_universitas` float NOT NULL,
  `bobot_total` int(2) NOT NULL,
  `jumlah_dokumen` int(2) NOT NULL,
  `tanggal_akses` datetime NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=484
DEFAULT CHARSET=latin1;
CREATE TABLE `tampung_akreditasi` (
  `id` int(2) unsigned NOT NULL
AUTO_INCREMENT,
  `nomor` int(3) NOT NULL,
  `nama_universitas` varchar(255) NOT NULL,
  `peringkat` varchar(150) NOT NULL,
  `masa_berlaku` varchar(150) NOT NULL,
  `status` varchar(150) NOT NULL,
  `skor` int(2) NOT NULL,
  `tmp_nama_universitas` text NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=13095
DEFAULT CHARSET=latin1;

CREATE TABLE `jumlah_mahasiswa` (

```

```

`id` int(2) unsigned NOT NULL
AUTO_INCREMENT,
`kode_universitas` int(4) NOT NULL,
`nama_universitas` varchar(255) NOT NULL,
`jumlah_tahun1_semester1` int(2) NOT NULL,

`jumlah_tahun2_semester1` int(2) NOT NULL,
`jumlah_tahun2_semester2` int(2) NOT NULL,
`status_universitas` varchar(50) NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=42783
DEFAULT CHARSET=latin1;

```

CREATE TABLE

```

`jumlah_tenaga_pengajar_jabatan_akademik` (
`id` int(2) NOT NULL AUTO_INCREMENT,
`kode_universitas` int(2) NOT NULL,
`nama_universitas` varchar(100) NOT NULL,
`asisten_ahli_10000` int(2) NOT NULL,
`asisten_ahli_15000` int(2) NOT NULL,
`lektor_20000` int(2) NOT NULL,
`lektor_30000` int(2) NOT NULL,
`lektor_kepala_40000` int(2) NOT NULL,
`lektor_kepala_55000` int(2) NOT NULL,
`lektor_kepala_70000` int(2) NOT NULL,
`profesor_85000` int(2) NOT NULL,
`profesor_105000` int(2) NOT NULL,
`tanpa_jabatan` int(2) NOT NULL,
`status_universitas` tinytext NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=16840 DEFAULT
CHARSET=latin1;

```

```

CREATE TABLE `jumlah_tenaga_pengajar_tetap` (
`id` int(2) NOT NULL AUTO_INCREMENT,
`kode_universitas` int(2) NOT NULL,
`nama_universitas` varchar(100) NOT NULL,

```

```

`D1` int(2) NOT NULL,
`D2` int(2) NOT NULL,
`D3` int(2) NOT NULL,
`D4` int(2) NOT NULL,
`S1` int(2) NOT NULL,
`S2` int(2) NOT NULL,
`S3` int(2) NOT NULL,
`non_formal` int(2) NOT NULL,
`informal` int(2) NOT NULL,
`lainnya` int(2) DEFAULT NULL,
`SP1` int(2) NOT NULL,
`SP2` int(2) NOT NULL,
`profesi` int(2) NOT NULL,
`tanpa_jenjang` int(2) NOT NULL,
`status_universitas` tinytext NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=59738 DEFAULT
CHARSET=latin1;
CREATE TABLE
`jumlah_tenaga_pengajar_tidak_tetap` (
  `id` int(2) NOT NULL AUTO_INCREMENT,
  `kode_universitas` int(2) NOT NULL,
  `nama_universitas` varchar(100) NOT NULL,
  `D1` int(2) NOT NULL,
  `D2` int(2) NOT NULL,
  `D3` int(2) NOT NULL,
  `D4` int(2) NOT NULL,
  `S1` int(2) NOT NULL,

  `S2` int(2) NOT NULL,
  `S3` int(2) NOT NULL,
  `non_formal` int(2) NOT NULL,
  `informal` int(2) NOT NULL,
  `lainnya` int(2) DEFAULT NULL,
  `SP1` int(2) NOT NULL,
  `SP2` int(2) NOT NULL,
  `profesi` int(2) NOT NULL,
  `tanpa_jenjang` int(2) NOT NULL,
  `status_universitas` tinytext NOT NULL,
PRIMARY KEY (`id`)

```

```

) ENGINE=InnoDB AUTO_INCREMENT=8885 DEFAULT
CHARSET=latin1;
CREATE TABLE `jumlah_sitasi_sinta` (
  `id` int(2) unsigned NOT NULL
  AUTO_INCREMENT,
  `nama_universitas` varchar(255) NOT NULL,
  `jumlah_author_scholar` int(2) NOT NULL,
  `jumlah_paper_scholar` int(2) NOT NULL,
  `jumlah_sitasi_scholar` int(2) NOT NULL,
  `jumlah_jurnal_artikel` int(2) NOT NULL,
  `jumlah_book_chapter` int(2) NOT NULL,
  `jumlah_paper_conference` int(2) NOT NULL,
  `jumlah_artikel_scopus` int(2) NOT NULL,
  `jumlah_non_artikel_scopus` int(2) NOT NULL,
  `jumlah_sitasi_scopus` int(2) NOT NULL,
  `sinta_score` int(2) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=4006
DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `jumlah_author_sinta` (
  `id` int(2) unsigned NOT NULL
  AUTO_INCREMENT,
  `id_data` int(2) NOT NULL,
  `nomor` int(2) NOT NULL,
  `nama_author` varchar(150) NOT NULL,
  `nama_universitas` varchar(150) NOT NULL,
  `jumlah_sitasi_scholar` int(2) NOT NULL,
  `i10_index_scholar` int(2) NOT NULL,
  `h_index_scholar` int(2) NOT NULL,
  `waktu_akses` tinytext NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2189 DEFAULT
CHARSET=latin1;

```

```

CREATE TABLE `precense` (
  `id` int(2) unsigned NOT NULL
  AUTO_INCREMENT,

```

```

`url_web` varchar(255) NOT NULL,
`domain_search` varchar(150) NOT NULL,
`hasil_search` int(2) NOT NULL,
`tanggal_akses` datetime NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=526 DEFAULT
CHARSET=latin1;
CREATE TABLE `impact` (
`id` int(2) unsigned NOT NULL
AUTO_INCREMENT,
`url_web` varchar(255) NOT NULL,
`domain_web` varchar(150) NOT NULL,
`page_authority` int(2) NOT NULL,
`domain_authority` int(2) NOT NULL,
`backlink_eksternal` int(2) NOT NULL,
`tanggal_akses` datetime NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1960
DEFAULT CHARSET=latin1;

CREATE TABLE `alexa_rank` (
`id` int(2) unsigned NOT NULL
AUTO_INCREMENT,
`url_web` varchar(255) NOT NULL,
`rank_world` int(2) NOT NULL,
`rank_country` int(2) NOT NULL,
`country_name` varchar(150) NOT NULL,
`tanggal_akses` datetime NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=444
DEFAULT CHARSET=latin1;

CREATE TABLE `kumpulan_skor_final` (
`id` int(2) unsigned NOT NULL
AUTO_INCREMENT,
`kode_universitas` int(3) NOT NULL,
`nama_universitas` varchar(255) NOT NULL,
`skor_reputasi_akademik_normalisasi` float
NOT NULL,

```

```

    `skor_nilai_akreditasi_normalisasi` float
NOT NULL,
    `skor_jumlah_mahasiswa_normalisasi` float
NOT NULL,
    `skor_jumlah_sitasi_normalisasi` float NOT
NULL,
    `skor_jumlah_tenaga_pengajar_normalisasi`
float NOT NULL,
    `skor_kualitas_sitasi_normalisasi` float NOT
NULL,
    `skor_jumlah_phd_normalisasi` float NOT NULL,
    `skor_jumlah_paper_normalisasi` float NOT
NULL,
    `skor_jumlah_jurnal_artikel` float NOT NULL,
    `skor_jumlah_paper_conference` float NOT
NULL,
    `skor_precense_normalisasi` float NOT NULL,
    `skor_impact_normalisasi` float NOT NULL,
    `skor_transparency_normalisasi` float NOT
NULL,
    `skor_excellence_normalisasi` float NOT
NULL,
    `skor_moz_analytic_normalisasi` float NOT
NULL,
    `skor_alex_rank_normalisasi` float NOT
NULL,
    `skor_kualitas_dosen_normalisasi` float NOT
NULL,
    `skor_kecukupan_dosen_normalisasi` float NOT
NULL,
    `skor_kualitas_manajemen_normalisasi` float
NOT NULL,
    `skor_kualitas_kegiatan_kemahasiswaan_normalis
asi` float NOT NULL,

    `skor_kualitas_kegiatan_penelitian_normalisasi`
    ` float NOT NULL,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=11356 DEFAULT
CHARSET=latin1;

```

```

CREATE TABLE `4icu` (
  `id` int(2) unsigned NOT NULL
  AUTO_INCREMENT,
  `nama_universitas` varchar(255) NOT NULL,
  `skor_4icu` float NOT NULL,
  `peringkat_4icu` int(2) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2128
DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `dikti` (
  `id` int(2) unsigned NOT NULL
  AUTO_INCREMENT,
  `nama_universitas` varchar(255) NOT NULL,
  `skor_dikti` float NOT NULL,
  `peringkat_dikti` int(2) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1773 DEFAULT
CHARSET=latin1;

```

```

CREATE TABLE `qs_general` (
  `id` int(2) unsigned NOT NULL
  AUTO_INCREMENT,
  `nama_universitas` varchar(255) NOT NULL,
  `skor_qs` float NOT NULL,
  `peringkat_qs` int(2) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2838
DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `qs_asia` (
  `id` int(2) unsigned NOT NULL
  AUTO_INCREMENT,
  `nama_universitas` varchar(255) NOT NULL,
  `skor_qs_asia` float NOT NULL,
  `peringkat_qs_asia` int(2) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2838
DEFAULT CHARSET=latin1;
CREATE TABLE `qs_llda` (

```

```

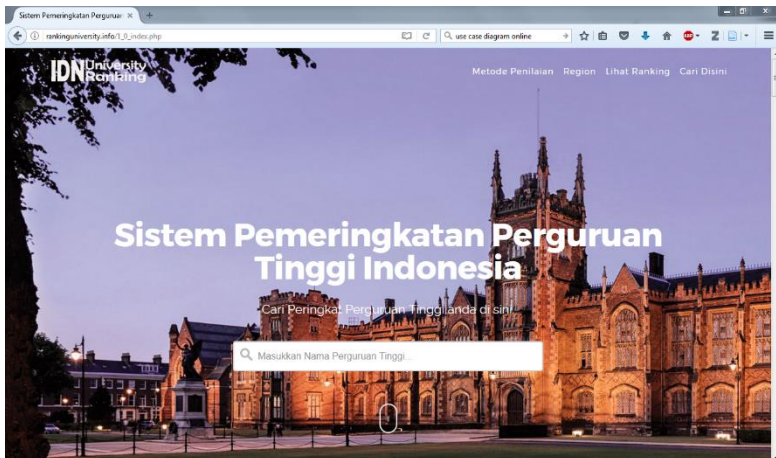
`id` int(2) unsigned NOT NULL
AUTO_INCREMENT,
`nama_universitas` varchar(255) NOT NULL,
`skor_qs_llda` float NOT NULL,
`peringkat_qs_llda` int(2) NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3193
DEFAULT CHARSET=latin1;

CREATE TABLE `webometrics` (
`id` int(2) unsigned NOT NULL
AUTO_INCREMENT,
`nama_universitas` varchar(255) NOT NULL,
`skor_webometrics` float NOT NULL,
`peringkat_webometrics` int(2) NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2838
DEFAULT CHARSET=latin1;

```

Kode Sumber A.5 Implementasi Basis Data

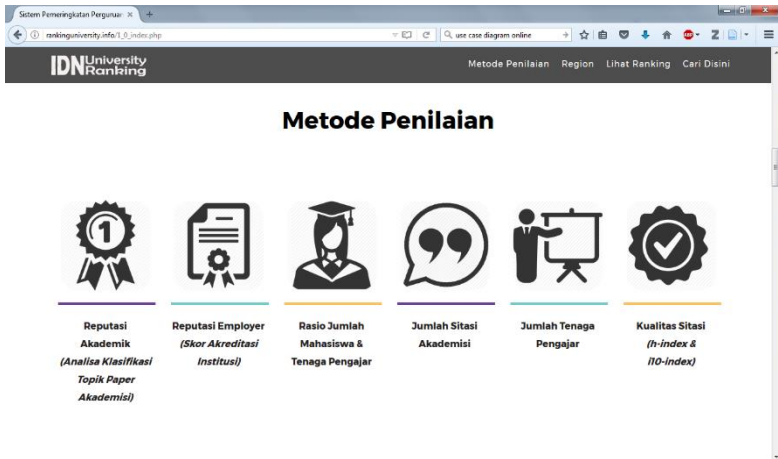
Gambar



Gambar A.1 Halaman Utama Bagian Pencarian



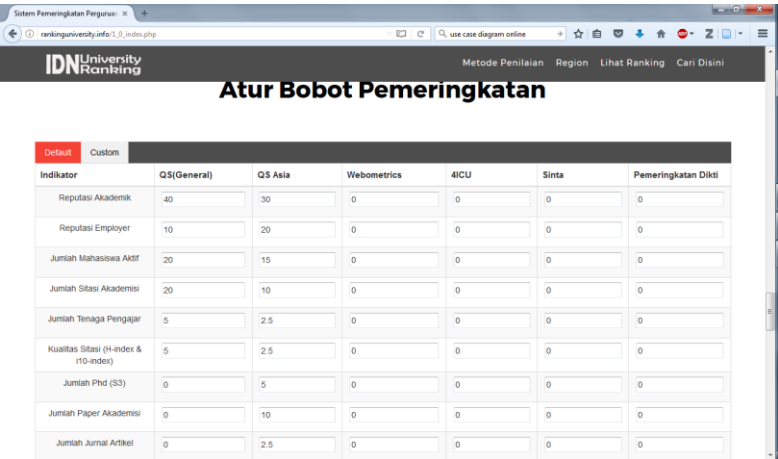
Gambar A.2 Halaman Utama Bagian Informasi Pemeringkatan



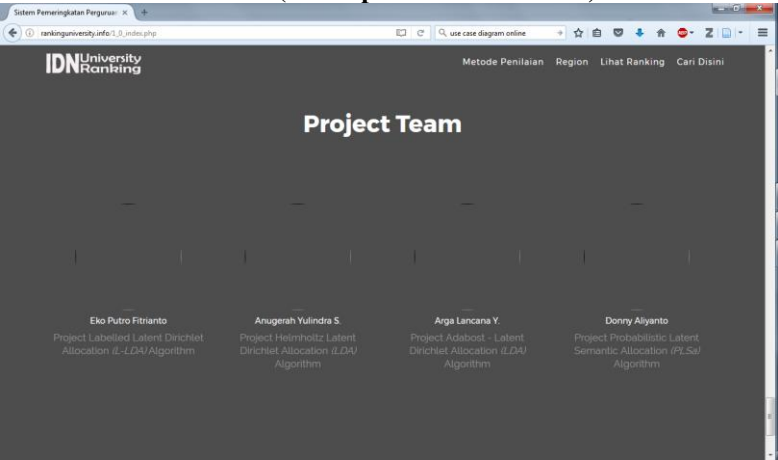
Gambar A.3 Halaman Utama Bagian Metode Penilaian



Gambar A.4 Halaman Utama Bagian Peta (Peringkat berdasarkan Region)



Gambar A.5 Halaman Utama Bagian Lihat Peringkat dan *tab* Custom (untuk pembobotan sendiri)



Gambar A.6 Halaman Utama Bagian Tentang Kami

Miniatur QS University Ranking

rankinguniversity.info/10_3_starch.php

Hasil Pencarian

No.	Nama Perguruan Tinggi	Kota	Provinsi	QS General	QS Asia	Webometrics	4ICU	Dikti	Lainnya
1	Institut Teknologi Sepuluh Nopember	Surabaya	Jawa Timur	Peringkat 8	Peringkat 7	Peringkat 6	Peringkat 12	Peringkat 3	Lihat Detail

Connect with us on [Twitter](#) [Facebook](#) [Instagram](#)

Copyright © 2016. All rights reserved.
Created by Designcrazed

Gambar A.7 Halaman Hasil Pencarian

Miniatur QS University Ranking

rankinguniversity.info/10_2_show_detail.php?u=Institut Teknologi Sepuluh Nopember

Institut Teknologi Sepuluh Nopember
Surabaya, Jawa Timur

Kepopuleran Website:

URL Website www.its.ac.id
Website dalam Ranking Dunia **26740**
Website dalam Ranking Nasional **476**
Page Authority **47**
Domain Authority **49**

Mahasiswa:
Mahasiswa Aktif **39223**

Status Akreditasi:
Nilai Akreditasi **A**
Tanggal Masa Berlaku **2019-01-16**
Status Masa Berlaku **masih berlaku**

Produktifitas Institusi:
Author Google Scholar **946**
Paper Google Scholar **2963**
Sitasi Google Scholar **19446**
Artikel Scopus **786**
Sitasi Scopus **6289**

Kecukupan Dosen:
Dosen Aktif **994**
Dosen Status Tetap **984**

Kualitas Adakemisi:

Gambar A.8 Halaman Detail Informasi Universitas

Miniatur QS University Ranking

rankinguniversity.info/2.1_view_data_qi_asia.php?x=0.38b+0.28c+0.158d+0.18e+0.0258f+0.0258g+0.0258h+0

use case diagram online

Peringkat Perguruan Tinggi (QS Asia)

No.	Nama Perguruan Tinggi	Skor	Kota	Provinsi
1	Institut Teknologi Bandung	0.7626	Bandung	Jawa Barat
2	Universitas Indonesia	0.7286	Depok	Jawa Barat
3	Universitas Gadjah Mada	0.7235	Yogyakarta	DI Yogyakarta
4	Institut Pertanian Bogor	0.6598	Bogor	Jawa Barat
5	Universitas Brawijaya	0.6165	Malang	Jawa Timur
6	Universitas Diponegoro	0.6053	Semarang	Jawa Tengah
7	Institut Teknologi Sepuluh Nopember	0.5995	Surabaya	Jawa Timur
8	Universitas Airlangga	0.5992	Surabaya	Jawa Timur
9	Universitas Hasanuddin	0.5671	Makassar	Sulawesi Selatan
10	Universitas Padjadjaran	0.5656	Bandung	Jawa Barat
11	Universitas Bina Nusantara	0.5471	Jakarta	DKI Jakarta

Gambar A.9 Halaman Hasil Pemeringkatan

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Arga Lancana Yuananda lahir di Ngawi pada tanggal 11 Mei 1995. Biasa dipanggil Arga, penulis saat ini sedang menempuh pendidikan Sarjana Teknik Informatika angkatan 2013 di Institut Teknologi Sepuluh Nopember dan memiliki seorang adik perempuan yang bersekolah di SMA Muhammadiyah Sidoarjo.

Memiliki ketertarikan dalam bidang perdagangan dan pernah menjadi reseller salah satu merk baju yang sedang *hype* di kalangan anak muda Indonesia. Terlibat aktif dalam organisasi kemahasiswaan serta kepanitiaan selama perkuliahan, antara lain staff Riset dan Teknologi di Himpunan Mahasiswa Teknik Computer-Informatika ITS 2014/2015 Staff Kesekretariatan acara Schematics 2014, dan Staff Web Kesekretariatan acara Schematics 2015. Untuk komunikasi, dapat dihubungi melalui surel : arga_lancana@yahoo.co.id.